

– **Dissertation** –

Ein Beitrag zur Pixel-basierten Verteilten Videocodierung
– Seiteninformationsgenerierung, WZ-Codierung und flexible Decodierung –

Ralph Hänsel

Rostock, den 23. Oktober 2013

Ein Beitrag zur Pixel-basierten
Verteilten Videocodierung
– Seiteninformationsgenerierung, WZ-Codierung
und flexible Decodierung –

Dissertation

zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

der Fakultät für Informatik und Elektrotechnik
der Universität Rostock

vorgelegt von
Ralph Hänsel, geboren am 12.Juli 1980 in Zerbst
aus Rostock

Rostock, den 23. Oktober 2013

Tag der Einreichung: 9. November 2012

Tag der öffentlichen Verteidigung: 20. März 2013

Gutachter:

Prof. Dr.-Ing. habil. Erika Müller

Universität Rostock, Institut für Nachrichtentechnik

Prof. Dr.-Ing. habil. Volker Kühn

Universität Rostock, Institut für Nachrichtentechnik

Prof. Dr.-Ing. André Kaup

Friedrich-Alexander Universität Erlangen Nürnberg,

Lehrstuhl für Multimediakommunikation und Signalverarbeitung

Dank an diejenigen,
die zum Gelingen dieser Arbeit beigetragen haben.

Kurzfassung

Eine der wesentlichen Aufgaben der Nachrichtentechnik stellt die Übertragung von audiovisuellen Medien dar. Der Rundfunk (Fernsehen, Radio) zählt zu den typischen Anwendungsszenarien. In modernen digitalen Übertragungssystemen werden konventionelle Videocodierungsverfahren eingesetzt. Diese sind aufgrund ihrer geringen Decoderkomplexität sehr gut für Rundfunkszenarien geeignet.

Neue individuellere Anwendungsszenarien, wie die Übertragung von mobilen Endgeräten, stellen neue Herausforderungen, wie eine geringe Encoderkomplexität, an das Videoübertragungssystem. Die Verteilte Videocodierung bietet die Möglichkeit, Systeme mit sehr geringer Encoderkomplexität zu entwerfen.

Die vorliegende Arbeit liefert einen Beitrag zur Verteilten Videocodierung. Im Fokus liegen die sehr geringe Encoderkomplexität sowie auch die Steigerung der Leistungsfähigkeit und die Verbesserung der Flexibilität des Decodierungsprozesses. Die sehr geringe Komplexität wird durch konsequente Umsetzung eines pixelbasierten Ansatzes sichergestellt.

Einer der wesentlichen Beiträge der Arbeit bezieht sich auf die Verbesserung der Seiteninformationsqualität durch temporale Interpolation. Hier werden insbesondere Probleme, wie sehr schnelle und inhomogene Bewegungen und Auf- und Verdeckungen, durch die Berücksichtigung von markanten Bildbereichen, gelöst.

Weiterhin wird die Leistungsfähigkeit stark durch die Wyner-Ziv Codierung beeinflusst. Diese wurde durch die leistungsfähige Turbo-Codierung umgesetzt. Die Leistungsfähigkeit der Codierung kann insbesondere durch Berücksichtigung ungleichmäßig verteilter Symbole, im Rahmen der Turbo-Codierung, erhöht werden.

In Verteilten Videocodierungssystemen stellt der Rückkanal ein wesentliches Hemmnis für viele Anwendungsszenarien dar. Die ansatzweise entwickelte flexible Decodierung gibt dem Decoder mehr Freiheitsgrade, um die Eliminierung des Rückkanals zu erleichtern.

Abschließend enthält die Arbeit eine Evaluation der entwickelten Verfahren. Der Vergleich mit konventionellen Videocodierungsverfahren hat gezeigt, dass für Sequenzen mit moderater Bewegungsaktivität eine mit H.264intra vergleichbare Leistungsfähigkeit erreicht werden konnte.

Abstract

The transmission of audiovisual media is one of the major tasks of communications engineering. Broadcasting (television and radio) is a typical application scenario. In modern digital transmission systems conventional video coding methods are used. Those coding methods are very suitable for the broadcasting scenario, due to their low decoding complexity.

New application scenarios such as video encoding on mobile devices pose new challenges. Mobile devices are very limited in computing capability. Thus the demand for low complexity video encoders is born. The distributed video coding theory provides methods to design video communication systems with very low encoding complexity.

This thesis is a contribution to the distributed video coding. It is focused on very low encoding complexity, increased rate distortion performance and improved decoding flexibility. The very low encoding complexity is ensured by following the pixel-domain coding approach.

One major contributions relates to the side information improvement by temporal interpolation. Major problems such as very fast and inhomogeneous motion as well as occlusions and revelation are considered. Local image features are incorporated to get a rough and robust estimation of the motion in a video sequence.

Furthermore, the rate distortion performance is strongly influenced by the Wyner-Ziv coding. It is implemented using the powerful turbo coding approach. The coding efficiency is particularly improved by modifying the turbo decoding process for non-uniform sources.

At least, the feedback channel is a major drawback of distributed video coding systems compared to conventional systems. Many application scenarios do not support a feedback channel. Flexible decoding is proposed, which is one step in the solution of the feedback channel problem. It provides a higher degree of freedom for the decoder, which can handle the received data in a more flexible way.

Finally, the evaluation of the developed methods shows equal performance compared to conventional video coding (e.g. H.264intra), for sequences with moderate motion.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Thematischer Hintergrund	1
1.2. Motivation und Zielsetzung	2
1.3. Gliederung der Arbeit	4
2. Grundlagen	7
2.1. Codierung	7
2.1.1. Quellencodierung	9
2.1.2. Kanalcodierung	11
2.2. Theoretische Grundlagen der Verteilten Quellencodierung	17
2.2.1. Klassifikation der Forschungsrichtungen	17
2.2.2. Verteilte Quellencodierung	19
2.3. Anwendungsszenarien für die Verteilte Videocodierung	21
2.4. Verteilte Videocodierungssysteme mit geringer Komplexität	23
2.4.1. Überblick: Verteilte Videocodierungssysteme mit geringer Encoderkomplexität	24
2.4.2. Verfahren zur Generierung der Seiteninformation	26
2.4.3. Wyner-Ziv und Slepian-Wolf Codierung	29
2.4.4. Modellierung	31
2.4.5. Zusammenfassung	32
3. Rostock Distributed Video Codec – <i>ρDeViCe</i>	33
3.1. Anforderungen und Zielsetzungen	33
3.2. Einfluss der DCT auf die Komplexität eines DVC-Encoders	34
3.2.1. Komplexität der DCT	34
3.2.2. Komplexität eines DVC Encoders	35
3.2.3. Fazit	37

3.3. Grundstruktur und Komponenten	38
4. Generierung der Seiteninformation – $\rho DeViCe$	41
4.1. Ausgewählte bekannte Verfahren zur Generierung der Seiteninformation	42
4.1.1. BiMESS – <i>Bidirectional Motion Estimation and Spatial Smoothing</i> & MCTI – <i>Motion Compensated Temporal Interpolation</i>	43
4.1.2. PBTI – <i>Pixel-based Temporal Interpolation</i>	44
4.1.3. Bewertung der bekannten Verfahren	46
4.2. Übersicht der entwickelten Verfahren zur Generierung der Seiteninformation	46
4.3. Vereinfachte pixelbasierte temporale Interpolation (basicPBTI) . . .	47
4.4. Globale Bewegung	51
4.4.1. Schätzung der globalen Bewegung	53
4.4.2. Adaptive Inter-/Extrapolation, unterstützt durch globale Bewegung (GTIE)	57
4.4.3. Pixelbasierte adaptive Inter-/Extrapolation, unterstützt durch globale Bewegung (GpTIE)	60
4.4.4. Vergleich der entwickelten Verfahren	62
4.4.5. Fazit	63
4.5. Inhomogene Bewegung	65
4.5.1. Lokale Bewegungshypothese	66
4.5.2. Multihypothesen pixelbasierte temporale Interpolation (MPBTI)	68
4.5.3. Vergleich von MPBTI mit bisherigen Verfahren	71
4.5.4. Fazit	73
4.6. Aufdeckung und Verdeckung	74
4.6.1. Adaptive pixelbasierte zeitliche Interpolation (APBTI) . . .	76
4.6.2. Erweiterte Adaptive pixelbasierte zeitliche Interpolation (APBTI2)	78
4.6.3. Vergleich von APBTI und APBTI2 mit den bekannten Verfahren	81
4.6.4. Fazit	82
4.7. Komplexität	83
4.7.1. Einflussfaktoren auf die Komplexität	83

4.7.2.	Ganzzahlige Koeffizienten zur Bestimmung des gewichteten SAD	86
4.7.3.	Reduktion der Dichte des Bewegungsvektorfeldes (Vorwärtsbewegungsschätzung)	88
4.7.4.	Hypothesenbasierte bidirektionale Bewegungsschätzung . . .	90
4.7.5.	Schnelle pixelbasierte temporale Interpolation (FPBTI) . . .	91
4.7.6.	Vergleich der schnellen pixelbasierten zeitlichen Interpolation (FPBTI) mit basicPBTI	92
4.7.7.	Fazit	94
4.8.	Zusammenfassung	94
5.	Wyner-Ziv Codierung – $\rho DeViCe$	97
5.1.	Slepian-Wolf Codierung	98
5.1.1.	Turbocodebasierte Slepian-Wolf Codierung	98
5.1.2.	Vergleich von LDPCA basierter und turbocodebasierter Slepian-Wolf Codierung	106
5.1.3.	Ungleichmäßig verteilte Quellensymbole	106
5.1.4.	Fazit	109
5.2.	Wyner-Ziv Codierung	110
5.2.1.	Wyner-Ziv System	111
5.2.2.	Kanalmodell	116
5.2.3.	Rate-Distortion Funktion und Rekonstruktion	120
5.2.4.	Ungleichmäßig verteilte Quellensymbole	123
5.2.5.	Fazit	125
5.3.	Einfluss des Verfahrens zur Codierung der Key-Frames auf das Gesamtsystem	126
5.3.1.	RD-Performanz der Key-Frame-Codecs	126
5.3.2.	Seiteninformationsqualität bei unterschiedlichen Key-Frame-Codecs	127
5.4.	Zusammenfassung	129
6.	Leistungsfähigkeit des Gesamtsystems – $\rho DeViCe$	131
6.1.	Untersuchungsstrategie	131
6.2.	Seiteninformationsqualität	132
6.2.1.	Mittlere Qualität der Seiteninformation	132
6.2.2.	Abhängigkeit von der Qualität der Key-Frames	134

6.2.3. Seiteninformationsqualität im Verlauf einer Sequenz	136
6.3. RD-Performanz	136
6.3.1. Codierungsparameter	136
6.3.2. Referenzverfahren	137
6.3.3. Einfluss des Key-Frame Codec und der Framerate	137
6.3.4. Einfluss der Seiteninformationsgenerierung	138
6.3.5. Schwankungen der Rekonstruktionsqualität	139
6.4. Zusammenfassung	141
7. Rückkanal in der Verteilten Videocodierung	143
7.1. Das Problem Rückkanal	144
7.2. Lösungsansätze	145
7.3. Anwendungsszenarien für die flexible Decodierung	148
7.4. Bitebenen übergreifende Decodierung	151
7.4.1. Lösungsansatz	151
7.4.2. Simulationsergebnisse	154
7.4.3. Fazit	155
7.5. Soft-Rekonstruktion	156
7.6. Fehlerlokalisierung und -verschleierung	158
7.6.1. Unschärfe Bestimmung der Fehlerposition	160
7.6.2. Verschleierung von Fehlern	161
7.6.3. Fazit	162
7.7. Zusammenfassung	162
8. Zusammenfassung und Ausblick	165
8.1. Zusammenfassung	165
8.2. Ausblick	167
Literaturverzeichnis	169

Anhang	193
A. Vertiefende Ausführungen zur Kanalcodierung und Bildverarbeitung	195
A.1. Kanalcodierung	195
A.1.1. LDPC - Low Density Parity Check	195
A.1.2. CRC - Cyclic Redundancy Check	197
A.2. Bildverarbeitung	198
A.2.1. Bewegungsschätzung	198
A.2.2. Merkmalsbereiche in Bildern	205
B. Videosequenzen	213
B.1. QCIF	213
B.2. CIF	216
B.3. SIF (NTSC)	216
C. Seiteninformation	217
C.1. Mittlere Qualität der Seiteninformation	217
C.2. Qualität der Seiteninformation per Frame	226
C.3. Differenz PSNR-Y Key-Frames und Seiteninformation	235
D. RD-Performanz des Gesamtsystems	245
D.1. Parameter	245
D.2. Rate-Distortion Punkte	246
D.3. Referenzen	246
D.4. RD-Performanz	248
Verwendete Abkürzungen	267
Verwendete Formelzeichen	269
Thesen	275

1

Kapitel 1

Einleitung

1.1. Thematischer Hintergrund

Die Übertragung von Medien, wie Video oder Audio, stellt eine der wesentlichen Aufgaben der Nachrichtentechnik dar. Typische Anwendungsszenarien sind zum einen der Rundfunk (Radio oder Fernsehen), sowie Telefonie als auch Videokonferenzen. Darüber hinaus gewinnt die individuelle Kommunikation zunehmend an Bedeutung.

Die anfangs analogen Übertragungsverfahren, zum Beispiel im Rundfunk oder der Telefonie, wurden später durch digitale Übertragungsverfahren ersetzt. Diese ermöglichen eine bessere Nutzung des zur Verfügung stehenden Übertragungskanals, wobei eine Trennung zwischen der Quellencodierung zur Reduktion der Datenmenge und der Kanalcodierung zum Fehlerschutz sowie der Anpassung an das Übertragungsmedium stattfindet.

Für Rundfunkdienste eignen sich Verfahren, wie zum Beispiel MPEG-2, MPEG-4, H.264, sehr gut. Aufgrund ihrer hohen Leistungsfähigkeit und geringen Komplexität des Decoders sind diese besonders für Szenarien mit einmaliger zentraler Encodierung sowie Decodierung an vielen Empfängern geeignet.

Die videobasierte Kommunikation zwischen mobilen Endgeräten oder die Umsetzung von Videosensornetzwerken stellt neue Herausforderungen, wie geringe Komplexität des Encoders oder auch Fehlerrobustheit, an das Videocodierungsverfahren.

Die sehr leistungsfähigen konventionellen Videocodierungsverfahren (z.B. MPEG-2, MPEG-4 und H.264) setzen am Encoder eine Bewegungsschätzung und -kompensation ein, welche wesentlich zur Performanz beiträgt. Darüber hinaus ist dies auch der wesentliche Grund für die hohe Komplexität des Encoders, jedoch sehr geringer Komplexität des Decoders.

Seit dem letzten Jahrzehnt hat die Forschungsaktivität, im Bereich der Verteilten Codierung (DSC) als auch im Speziellen im Bereich der Verteilten Videocodierung (DVC), stark zugenommen. Ziel der Forschungsbestrebungen war es auch, Encoder-/Decoderstrukturen zu entwickeln, welche über eine geringe Komplexität des Encoders verfügen. Dies wird durch eine Verlagerung der komplexen Bewegungsschätzung von der Encoder- auf die Decoderseite erreicht.

Nach den Prinzipien der Verteilten Codierung werden Symbole von korrelierten Quellen getrennt voneinander codiert. Die Decodierung wird hingegen zentral durchgeführt, wodurch die Auswertung von Redundanzen möglich wird und somit die Leistungsfähigkeit der Quellencodierung gesteigert werden kann. Man spricht hierbei von Intra-Encodierung und Inter-Decodierung.

1.2. Motivation und Zielsetzung

Verteilte Videocodierungssysteme können das Anwendungsszenario der Encoder mit geringer Komplexität abdecken, wobei theoretisch [Wyner u. Ziv 1976] keine Reduktion der Leistungsfähigkeit in Vergleich zu konventionellen Systemen zu erwarten ist. Derzeit bekannte Codierungssysteme haben jedoch gezeigt, dass zum Teil große Lücken zwischen der Leistungsfähigkeit verteilter und konventioneller Videocodierungsverfahren bestehen. Diese Herausforderung stellt eine grundlegende Motivation für die vorliegende Arbeit dar.

Ein wesentlicher Schritt zur Verbesserung der Performanz stellte die Einführung der diskreten Cosinus-Transformation (DCT) in bekannten Verteilten Videocodierungssystemen dar. Dies führt jedoch zu einer erhöhten Encoderkomplexität, was dem Anwendungsziel, einen Encoder mit geringer Komplexität zu entwickeln, entgegen läuft. Aus diesem Grund ist diese Arbeit auf die Entwicklung eines pixel-basierten Encoder-/Decoder Systems gerichtet, da diese Systeme eine sehr geringe Encoderkomplexität aufweisen.

Die Verbesserung der Leistungsfähigkeit soll im Wesentlichen durch die Steigerung der Qualität der Seiteninformation und der Performanz der Slepian-Wolf (SW) Codierung erreicht werden. Ziel der Arbeit ist es, eine leistungsfähige Seiteninformationsgenerierung zu entwickeln und weiterhin die notwendigen Quellencodierungsprozesse zu optimieren.

Eine hohe Qualität der Seiteninformation lässt sich durch temporale Interpolation erreichen. Die zuverlässige Bestimmung der wahren Bewegung ist hierfür Voraussetzung und führt in Sequenzen mit komplexen Bewegungsstrukturen zu Problemen. Diese Herausforderung wird im Rahmen der Arbeit durch einen zweistufigen Ansatz, bestehend aus robuster Abschätzung der Bewegung und verfeinernder Suche, gelöst. Zur robusten Abschätzung der Bewegung werden insbesondere markante Bildbereiche einbezogen. Darüber hinaus wird die temporale Extrapolation genutzt, sofern dies zielführend ist.

Die Komplexität der temporalen Interpolation wird darüber hinaus durch zielgerichtete Berücksichtigung der markanten Bildbereiche reduziert.

Die Flexibilität von Verteilten Videocodierungssystemen ist im Vergleich zu konventionellen Systemen durch den Rückkanal stark eingeschränkt. Der Rückkanal ist notwendig, um eine effiziente Ratenkontrolle durchzuführen, setzt jedoch eine sofortige Decodierung voraus und verhindert somit die Speicherung der codierten Videosequenzen. Dieses Problem hat in der Forschungsgemeinschaft nur wenig Aufmerksamkeit erhalten, woraus sich der zweite Teil der Arbeit ableitet. Die Erhöhung der Flexibilität des Decoders durch Verfahren, welche es erlauben, den Rückkanal nur teilweise zu nutzen oder sogar gänzlich zu vermeiden, steht hier im Vordergrund.

Eine höhere Flexibilität des Decoders kann zum Beispiel erreicht werden, indem dieser die empfangenen Daten umschichten kann, um so eine erfolgreiche Decodierung zu ermöglichen. Darüber hinaus stellt sich die Frage, in wieweit Ansätze aus der Fehlerverschleierung geeignet sind, um die Rekonstruktionsqualität, im Fall eines Decodiersversagens, zu erhöhen.

1.3. Gliederung der Arbeit

Die vorliegende Arbeit ist in acht Kapitel unterteilt, in denen zum einen die Entwicklung des Verteilten Videocodierungssystems $\rho DeViCe$ im Detail beschrieben wird und zum anderen auf die Problematik des Rückkanals eingegangen wird. Die Zuordnung der Kapitel und Komponenten zueinander ist in Abbildung 1.1 dargestellt.

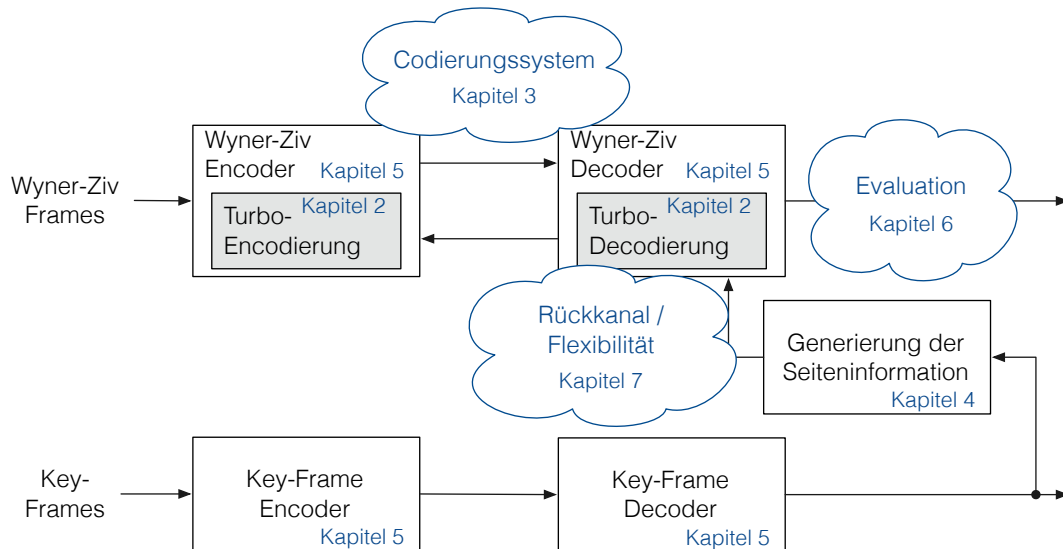


Abbildung 1.1.: Struktur eines Verteilten Videocodierungssystems mit Zuordnung zu den entsprechenden Kapiteln dieser Arbeit

Kapitel 2 (S. 7) — Das folgende Kapitel gibt die notwendigen Grundlagen zur Kanalcodierung wieder. Darüber hinaus erfolgt eine Einordnung des Forschungsgebietes der Verteilten Videocodierung und die Betrachtung der möglichen Anwendungsszenarien. Abschließend wird ein Überblick über bekannte Verfahren gegeben.

Kapitel 3 (S. 33) — Die Zielsetzung für den entwickelten Codec $\rho DeViCe$ wird in diesem Kapitel konkretisiert und aufbauend die Systemstruktur entworfen. Besondere Berücksichtigung erhält die Komplexität der DCT am Encoder.

Kapitel 4 (S. 41) — Die für die Leistungsfähigkeit mit entscheidende Generierung der Seiteninformation, unter Verwendung der temporalen Interpolation, wird in Kapitel 4 erarbeitet. Es wird insbesondere auf Sequenzen mit komplexen

Bewegungsstrukturen eingegangen und die Komplexität der entwickelten Verfahren berücksichtigt.

Kapitel 5 (S. 97) — Kapitel 5 beschreibt die Wyner-Ziv Codierung und die Umsetzung der Slepian-Wolf Codierung durch Turbocodes. Darüber hinaus wird die Codierung der Key-Frames in die Betrachtung des Gesamtsystems aufgenommen.

Kapitel 6 (S. 131) — Die Bewertung der Leistungsfähigkeit hinsichtlich der Qualität der Seiteninformation sowie hinsichtlich der Rate-Distortion Performanz des Gesamtsystems erfolgt in diesem Kapitel.

Kapitel 7 (S. 143) — Das Problem des Rückkanals in der Verteilten Videocodierung wird in Kapitel 7 erörtert. Weiterführend werden bekannte Lösungen bewertet und eigene Ansätze, basierend auf der Einteilung in Problemszenarien, entworfen.

Kapitel 8 (S. 165) — Abschließend erfolgt im Kapitel 8 eine Zusammenfassung der Ergebnisse sowie der Ausblick auf weitere offene Forschungsprobleme im Gebiet der Verteilten Videocodierung.

2

Kapitel 2

Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen der konventionellen Codierung dargelegt. Darüber hinaus erfolgt eine Klassifikation der Forschungsrichtungen zur Einordnung und Abgrenzung des Gebietes der Verteilten Quellencodierung. Die Anwendungsszenarien werden nachfolgend aufgezeigt und abschließend die aus der Literatur bekannten Verfahren kurz vorgestellt.

2.1. Codierung

Eine der wesentlichen menschlichen Fähigkeiten ist die Möglichkeit zur Kommunikation und somit zum Informationsaustausch. Der technische Nachrichtenaustausch ist dem natürlichen Vorbild nachempfunden. So besteht ein Kommunikationssystem aus einer Informationsquelle (Gedanke im Kopf, Daten in einem Informationsverarbeitungssystem) und einem Sender (Vokaltrakt, Funkmodem), welcher die Information in Signale passend zum Übertragungsmedium (Schall- bzw. elektromagnetische Welle in Luft) umwandelt. Bei der Übertragung treten Störeinflüsse auf, welche durch den Empfänger (Ohr, Funkmodem) kompensiert werden müssen, sodass die Information fehlerfrei zur Senke gelangt (Abb. 2.1).

Codierung beschreibt den Prozess der Umwandlung von Information in ein Signal zur Übertragung/Speicherung und die entsprechende Rekonstruktion der Information auf der Empfangsseite. Die Informationstheorie stellt dabei die mathematischen Grundlagen fachübergreifend zur Verfügung [Shannon 1948; Shannon u. Weaver

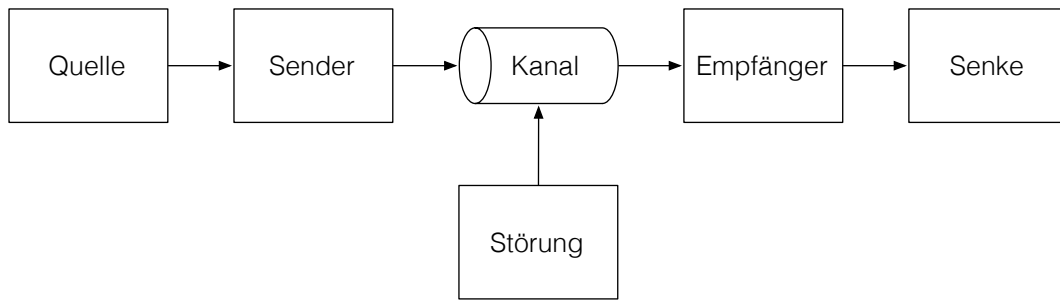


Abbildung 2.1.: Kommunikationssystem nach [Shannon 1948][Shannon u. Weaver 1949]

1949]. Gedanklich kann Codierung in die drei Komponenten Quellencodierung, Chiffrierung und Kanalcodierung zerlegt werden (sofern perfekte Quellen- und Kanalcodierung vorausgesetzt wird, [Höher 2011]).

Quellencodierung — Erzeugung einer effizienten Darstellung der Information, Entfernung von Redundanz, Reduktion des Datenumfanges

Chiffrierung — Verschlüsselung, Schutz der kommunizierten Information vor dem Zugriff Dritter

Kanalcodierung — Korrektur von Übertragungsfehlern und somit Anpassung an den Übertragungskanal zur fehlerrobusten Übertragung

Eine zentrale Bedeutung nimmt der Begriff **Information** ein, welche durch Nachrichten bzw. Signale übertragen wird und die Unbestimmtheit an der Senke verringert (Gl. 2.1). Im Sinne der Informationstheorie und aus Sicht des Systementwurfs ist die konkrete Nachricht aber nur von geringerem Interesse, wohingegen die Menge der möglichen Nachrichten von großer Bedeutung ist („... *information in communication theory relates not so much to what you do say, as to what you could say.*“, W.Weaver [Shannon u. Weaver 1949]). Ein System muss somit für den mittleren Informationsgehalt (Entropie, Gl. 2.2) entworfen werden.

$$I(X = x_\nu) = -\log_2 \Pr(X = x_\nu) \quad [\text{bit}] \quad (2.1)$$

$$H(X) = \sum_\nu \Pr(X = x_\nu) \cdot I(X = x_\nu) \quad [\text{bit/symbol}] \quad (2.2)$$

dabei gibt $\Pr(X = x_\nu)$ die Wahrscheinlichkeit an, dass die diskrete Quelle X das Ereignis x_ν ausgibt.

Weitere wichtige Informationsgrößen sind die Verbundinformation $H(X, Y)$ (Verbundentropie), die gemeinsame Information $I(X, Y)$ (Transinformation) und der bedingte Informationsgehalt $H(X|Y)$ (Abb. 2.2).

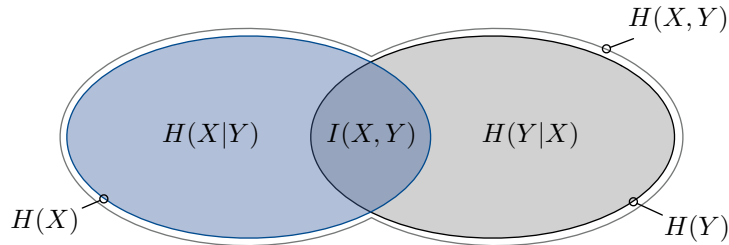


Abbildung 2.2.: Zusammenhang zwischen Entropie $H(X)$, $H(Y)$, Verbundentropie $H(X, Y)$, bedingte Entropie $H(X|Y)$, $H(Y|X)$ und Transinformation $I(X, Y)$

In den folgenden Abschnitten wird, beginnend mit der Quellencodierung (Abschn. 2.1.1), auch die Kanalcodierung (Abschn. 2.1.2) beschrieben. Der Bereich der Verteilten Quellencodierung (*distributed source coding*, DSC) wird gesondert in Abschnitt 2.2 behandelt. Weiterführende Ausführungen zur Kanalcodierung befinden sich darüber hinaus im Anhang A.

2.1.1. Quellencodierung

Die Quellencodierung hat die Aufgabe, die Redundanz zu verringern. Die Quellencodierung kann in zwei Gruppen, der verlustlosen sowie der verlustbehafteten Quellencodierung, eingeteilt werden [Cover u. Thomas 1991][Strutz 2002][Ohm 2004][MacKay 2005]. Im ersten Fall können nach der Decodierung die Originalsymbole der Quelle X zurückgewonnen werden, wobei im zweiten Fall nur eine fehlerbehaftete Approximation \hat{X} rekonstruiert werden kann.

Verlustlose Quellencodierung

Die notwendige Datenrate für die verlustlose Speicherung oder Übertragung richtet sich nach der Entropie der Quelle. Der Hauptsatz der Quellencodierung (Gl. 2.3)

gibt an, dass die Datenmenge pro Symbol nicht kleiner als die Entropie sein kann. Dabei seien große Mengen von Symbolen und statistisch unabhängige Symbole vorausgesetzt. Prominente Beispiele sind die Huffman-Codierung oder auch die Arithmetische Codierung.

$$R_X \geq H(X) \quad (2.3)$$

Verlustbehaftete Quellencodierung

Viele Codierungssysteme für Videosequenzen zählen zu der Gruppe der verlustbehafteten Quellencodierung, bei der keine fehlerfreie Rekonstruktion möglich ist. Damit wird eine geringe Datenrate erreicht, wodurch der praxistaugliche Einsatz möglich wird.

Der Zusammenhang zwischen minimaler Datenrate R und Verzerrung D (*distortion*, $d(x, \hat{x})$) wird durch die *Rate-Distortion* Funktion $R(D)$ angegeben (Gl. 2.4, [Cover u. Thomas 1991, S.336ff]).

$$R \geq R(D) = \min_{p_{\hat{X}|X}: \sum_{(x_j, \hat{x}_j)} p_X(x_j) p_{\hat{X}|X}(\hat{x}_j|x_j) d(x_j, \hat{x}_j) \leq D} I(X, \hat{X}) \quad (2.4)$$

In der Videocodierung wird die Verzerrung (*distortion*) als quadratischer Fehler bestimmt, welcher im Wesentlichen durch den Einsatz einer Quantisierung beeinflusst wird. Dies spiegelt sich in der Wahrscheinlichkeitsdichte $p_{\hat{X}|X}(\hat{x}_j|x_j)$ wieder (Gl. 2.4).

In natürlichen Videosequenzen oft auftretende örtliche und sehr starke zeitliche Abhängigkeiten werden durch örtliche Transformation bzw. Bewegungsschätzung und -kompensation aufgelöst. Diese leistungsfähige Prädiktion hat starken Einfluss auf die hohe Performanz aktueller Videocodierungsverfahren (H.264 [Wiegand u. a. 2003], HEVC [Bross u. a. 2012]) und spielt somit eine zentrale Rolle. Ein Überblick zu bekannten Verfahren der Bewegungsschätzung und -kompensation ist im Anhang A.2.1 (S. 198) zusammengestellt.

2.1.2. Kanalcodierung

Die Kanalcodierung hat das Ziel, einen gegebenen Informationsvektor \mathbf{i} mit möglichst geringer Fehlerrate über einen Kanal zu übertragen. Dazu wird jedem möglichen Informationsvektor $\mathbf{i}_{1 \times k}$ der Länge k ein Codevektor $\mathbf{c}_{1 \times n}$ der Länge $n > k$ zugeordnet (Abb. 2.3). Die damit hinzugefügte Redundanz wird quantitativ durch die Coderate R_c erfasst (Gl. 2.5). Im Bereich der Verteilten Quellencodierung (*Distributed Source Coding*) wird auch die Quellenrate (*source rate*) R_s verwendet, welche den Anteil der hinzugefügten Prüfsymbole im Vergleich zur Menge der Informationssymbole angibt.

$$R_c = \frac{k}{n} \quad R_s = \frac{n-k}{k} = \frac{1}{R_c} - 1 \quad (2.5)$$

$$\mathbf{i} = (i_0, i_1, \dots, i_{k-1}) \quad \mathbf{c} = (c_0, c_1, \dots, c_{n-1}) \quad (2.6)$$

Die Zuordnung eines binären Codevektors \mathbf{c} zu einem binären Informationsvektor \mathbf{i} erfolgt für die, in der Verteilten Quellencodierung relevanten, linearen Blockcodes durch Multiplikation von \mathbf{i} mit der Generatormatrix \mathbf{G} [Bossert 1998, S.22]. Dabei erfüllt jedes Codewort \mathbf{c} die Bedingung, dass sein Produkt mit der Prüfmatrix \mathbf{H} den Nullvektor ergibt (Gl. 2.7).

$$\mathbf{c}_{1 \times n} = \mathbf{i}_{1 \times k} \cdot \mathbf{G}_{k \times n} \quad \mathbf{H}_{n-k \times n} \cdot \mathbf{c}_{1 \times n}^T = \mathbf{0} \quad (2.7)$$

Für die Verteilte Quellencodierung werden auch die systematischen linearen Blockcodes eingesetzt, bei denen der Informationsvektor \mathbf{i} Teil des Codevektors \mathbf{c} , $i_j = c_j, j \in [0 \dots k-1]$ ist. Für die Generator- und Prüfmatrix ergibt sich damit die in Gleichung 2.8 angegebene Form. Die Teilmatrix \mathbf{A} gibt dabei an, wie die Prüfsymbole $c_k \dots c_{n-1}$ aus dem Informationsvektor \mathbf{i} gebildet werden. Im Fall von binären Symbolen i_j, c_j gilt: $\mathbf{A} = -\mathbf{A}$.

$$\mathbf{G} = (\mathbf{I}_k | -\mathbf{A}^T) \quad \mathbf{H} = (\mathbf{A} | \mathbf{I}_{n-k}) \quad (2.8)$$



Abbildung 2.3.: Kanalcodierung zum Schutz der Übertragung eines Informationsvektors \mathbf{i} über einen gestörten Kanal

Kanalmodelle

Zwei in der Übertragungstechnik verbreitete Kanalmodelle sind zum einen der symmetrische Binärkanal (BSC, *binary symmetric channel*) und zum anderen der AWGN-Kanal (*additive white Gaussian noise*, additives weißes Gauß-Rauschen). Das BSC-Kanalmodell besitzt sowohl einen binären Ein- sowie Ausgang und wird durch die Übergangswahrscheinlichkeit e für ein falsch übertragenes Bit charakterisiert (Abb. 2.4(a)). Dem hingegen werden die diskreten Codesymbole c_ν im AWGN-Kanalmodell (Abb. 2.4(b)) durch den Symbolformer auf kontinuierliche Kanaleingangssymbole x abgebildet. Diese werden mit weißem Gauß-Rauschen n überlagert. Der Kanalausgang y ist hierbei kontinuierlich und wird meist zur Bestimmung von Zuverlässigkeitsinformation direkt in der Decodierung des Kanalcodes verwendet.

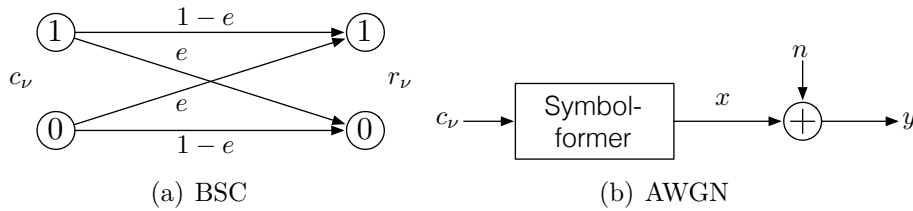


Abbildung 2.4.: Kanalmodelle

Die Transinformation für Kanäle mit diskreten Eingangssymbolen x_ν und kontinuierlichen Ausgangssymbolen y kann nach Gleichung 2.9 bestimmt werden ([Kühn 2006], S.59). Diese kann für viele Kanäle nicht analytisch gelöst werden, womit eine numerische Lösung zur Bestimmung der Transinformation herangezogen werden muss.

$$I(X; Y) = \sum_{\nu} \Pr(X = x_{\nu}) \int_y p_{Y|X}(y|x_{\nu}) \log_2 \frac{p_{Y|X}(y|x_{\nu})}{\sum_{\mu} \Pr(X = x_{\mu}) p_{Y|X}(y|x_{\mu})} dy \quad (2.9)$$

Für Kanäle mit diskreten Ein- sowie Ausgangssymbolen x_ν , y_ν kann die Transinformation nach Gleichung 2.10 bestimmt werden. Für den Fall gleichverteilter binärer Eingangssymbole vereinfacht sich die Bestimmung der Transinformation für den BSC-Kanal (Gl. 2.11). Für diesen Fall wird die maximale Transinformation für einen BSC-Kanal erreicht, was somit dessen Kanalkapazität C entspricht.

$$I(X; Y) = \sum_{\nu} \Pr(X = x_{\nu}) \sum_{\kappa} \Pr(Y = y_{\kappa} | X = x_{\nu}) \cdot \log_2 \frac{\Pr(Y = y_{\kappa} | X = x_{\nu})}{\sum_{\mu} \Pr(X = x_{\mu}) \Pr(Y = y_{\kappa} | X = x_{\mu})} \quad (2.10)$$

$$I(X; Y) = 1 + (1 - e) \log_2(1 - e) + e \log_2(e) \quad (2.11)$$

Turbo-Codierung

Der Begriff Turbocode ist vom Verfahren der Turbo-Decodierung für verkettete Kanalcodes (*concatenated codes*) abgeleitet. Dabei handelt es sich um ein iteratives Decodierungsverfahren von zwei oder mehr miteinander verketteten Codes. Faltungscodes werden oft miteinander verkettet, da der Gesamtcode eine hohe Fehlerkorrekturleistung und Flexibilität hinsichtlich der Coderate zeigt.

Faltungscodes zählen zu der Gruppe der linearen Codierungsverfahren, jedoch nicht zu den Blockcodes, da die Codewortlänge n variabel ist. Der binäre Codevektor \mathbf{c} wird aus dem Informationsvektor \mathbf{i} durch Anwendung eines zeitdiskreten LTI-Systems erzeugt [Bossert 1998, S.227ff].

Abbildung 2.5(a) zeigt einen systematischen rekursiven Faltungscodierer (RSC). Die Struktur des Filters (Codierers) wird im Frequenzbereich angegeben, wobei im Bereich der z-Transformation $D = z^{-1}$ gilt. Der systematische Anteil \mathbf{c}_i wird durch den Generator $g_1(D) = 1$ und die Paritätssymbole \mathbf{c}_p werden durch $g_2(D) = \frac{1+D+D^2+D^3}{1+D+D^3}$ gebildet. Die oktale Schreibweise der Form $\left[1, \frac{17}{15}\right]_8$ ist auch üblich. Dieser Code besitzt eine Gedächtnislänge (Einflusslänge) von $L_c = 4$ und eine Coderate von $R_c = \frac{1}{2}$.

Die Encodierung kann durch einen Zustandsgraphen ausgedrückt werden, wobei der Inhalt des Gedächtnisses den aktuellen Zustand $\mathbf{s} = 000 \dots 111$ angibt. Der zeitliche

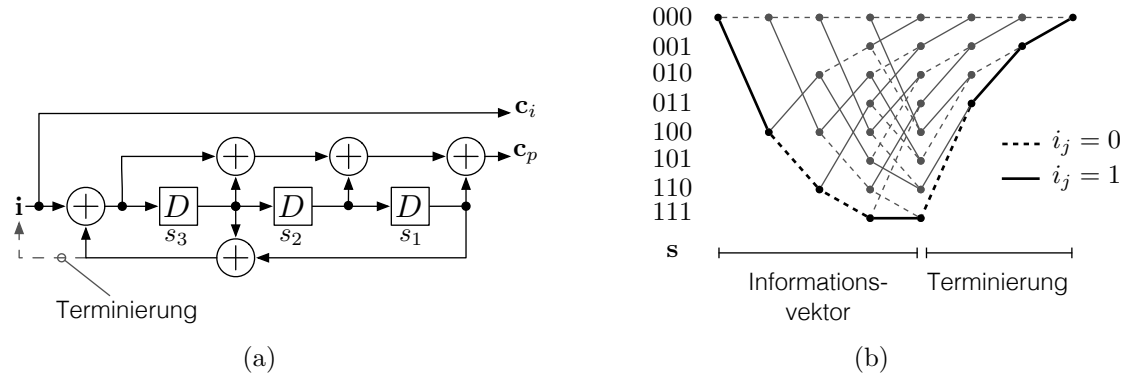


Abbildung 2.5.: (a) Rekursiver Faltungscodierer, $\left[1, \frac{17}{15}\right]_8$ und (b) zugehöriger Trellis für $\mathbf{i} = (1001)$, $\mathbf{c}_i = (1001011)$, $\mathbf{c}_p = (1010101)$

Ablauf der Zustandswechsel wird in einem Trellis (Abb. 2.5(b)) von links nach rechts dargestellt. Die schwarzen Linien stellen den Pfad für den Informationsvektor $\mathbf{i} = (1001)$ und den zugehörigen Terminierungssymbolen dar. Eine Terminierung ist sinnvoll, da so der Endzustand des Encoders $\mathbf{s} = 000$ ist und somit auch am Decoder bekannt ist. Damit kann die Leistungsfähigkeit der Decodierung erhöht werden.

Der Aufwand zur Encodierung eines halbratigen RSC Codes kann bei gegebener Gedächtnislänge L_c nach oben abgeschätzt werden, wobei $n_{add} \leq 2k(L_c - 1)$ Additionen (bzw. XOR-Verknüpfungen) zur Encodierung eines Codewortes der Länge k notwendig sind.

Die Verkettung einfacher Codes bietet wesentliche Steigerungen der Leistungsfähigkeit, da durch die Verkettung ein neuer komplexerer Code gebildet wird. Darüber hinaus bleibt die Komplexität der Decodierung gering, da Decoder für die einfachen Teilcodes eingesetzt werden, welche Informationen untereinander austauschen.

Die parallele Verkettung eines rekursiven Faltungscodes (RSC) wurde in [Berrou u. a. 1993] unter dem Namen *Turbo*code vorgeschlagen und erregte durch die sehr hohe Leistungsfähigkeit großes Forschungsinteresse. Turbo-Decodierung [Moreira u. Farrell 2006, S.209ff] [Kühn 2006, S.135ff] [MacKay 2005, S.574ff] kann auf seriell und parallel verkettete Codes angewendet werden, wobei hier nur die parallele Verkettung betrachtet wird.

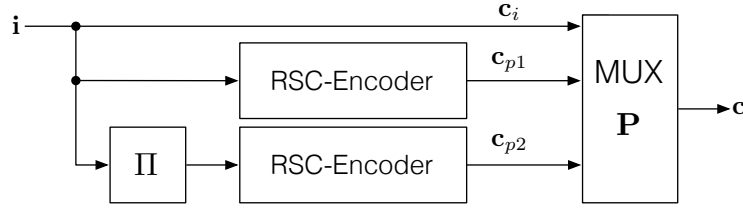


Abbildung 2.6.: Turbo-Encoder für einen parallel verketteten Faltungscode

Der vom **Encoder** (Abb. 2.6) generierte Codevektor \mathbf{c} setzt sich aus drei Teilen zusammen. Dies ist der systematische Anteil \mathbf{c}_i , der Teilcodevektor \mathbf{c}_{p1} aus dem RSC encodierten Informationsvektor \mathbf{i} , und dem Teilcodevektor \mathbf{c}_{p2} , wobei hier die Elemente des Informationsvektors \mathbf{i} zufällig vertauscht wurden (Interleaving, Π). Der Interleaver stellt dabei die Unabhängigkeit der Teilcodeworts sicher. Es hat sich gezeigt, dass ein langer Interleaver (großes k) zu einer besseren Leistungsfähigkeit führt.

Die parallele Verkettung zweier $\frac{1}{2}$ -ratiger RSC Codes führt zu einem Code mit Coderate von $R_c = \frac{1}{3}$. Für die verteilte Quellencodierung sind nur Coderaten nahe 1 von Interesse, da so eine sehr geringe Datenrate erreicht werden kann. Zur Erhöhung der Coderate werden einzelne Symbole aus dem Codevektor $\mathbf{c} = (\mathbf{c}_i | \mathbf{c}_{p1} | \mathbf{c}_{p2})$ entfernt. Jede Zeile der Punktierungsmatrix \mathbf{P} gibt für jeden Teilcodevektor an, welche Symbole übertragen ($p_{i,j} = 1$) oder punktiert ($p_{i,j} = 0$) werden (Gl. 2.12). Für das hier genutzte Beispiel hat die Punktierungsmatrix die Dimension $3 \times L_p$. Die Matrix \mathbf{P} wird mehrmals horizontal aneinandergefügt, um auch größere Codewortlängen n abzudecken (\mathbf{P}' , Gl. 2.13).

$$\mathbf{P} = \begin{pmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,L_p-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{2,0} & p_{2,1} & \cdots & p_{2,L_p-1} \end{pmatrix} \begin{array}{l} \leftarrow \mathbf{c}_i \\ \leftarrow \mathbf{c}_{p1} \\ \leftarrow \mathbf{c}_{p2} \end{array} \quad (2.12)$$

$$\mathbf{P}' = (\mathbf{P} | \mathbf{P} | \dots | \mathbf{P}) \quad (2.13)$$

Die Encodierung mit einem parallel verketteten Faltungscode der Rate $R_c = \frac{1}{3}$ benötigt $n_{add} \leq 4k(L_c - 1)$ Additionen. Für die parallele Verkettung des in Abbildung 2.5(a) dargestellten Encoders sind es $n_{add} = 10k$ ohne Terminierung.

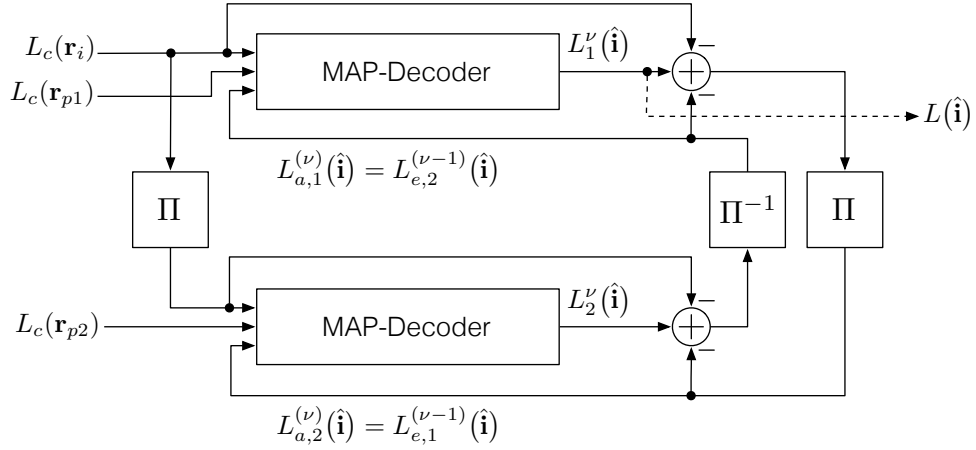


Abbildung 2.7.: Turbo-Decoder für einen parallel verketteten Faltungscode

Der **Decoder** für zwei parallel verkettete Faltungscode besteht aus zwei Maximum-a-posteriori Decodern (MAP) je für einen Faltungscode, welche iterativ Informationen austauschen (Abb. 2.7). Dabei werden Zuverlässigkeitsinformationen über jedes Symbol des Empfangsvektors \mathbf{r} verarbeitet, welche durch das *Log-Likelihood-Ratio* (LLR, $L_c(r_j) = \ln \frac{p_{R|C}(r_j|c_j=0)}{p_{R|C}(r_j|c_j=1)}$) abgebildet wird. Das Vorzeichen des LLR gibt dabei die harte Entscheidung und der Betrag gibt die Zuverlässigkeit an.

Ziel der Decodierung ist es, für jedes Symbol des Informationsvektors \mathbf{i} die Wahrscheinlichkeit zu bestimmen, ob eine 0 bzw. 1 gesendet wurde (posteriori LLR, Gl. 2.14). Der posteriori LLR lässt sich in drei Anteile zerlegen (Gl. 2.15). Der a-priori LLR L_a bildet die Eigenschaften der Quelle (Quellenentropie), der intrinsische LLR L_c die Eigenschaften des Kanals und der extrinsischen LLR L_e den Gewinn durch den Kanalcode ab.

$$L(\hat{i}_j) = \ln \frac{\Pr(i_j = 0|\mathbf{r})}{\Pr(i_j = 1|\mathbf{r})} \quad (2.14)$$

$$= L_a(\hat{i}_j) + L_c(\hat{i}_j) + L_e(\hat{i}_j) \quad (2.15)$$

$$= \ln \frac{\Pr(i_j = 0)}{\Pr(i_j = 1)} + \ln \frac{p_{R|I}(r_j|i_j = 0)}{p_{R|I}(r_j|i_j = 1)} + L_e(\hat{i}_j) \quad (2.16)$$

Während der iterativen Decodierung wird die durch einen MAP-Decoder gewonnene extrinsische Information in der nächsten Iteration ν von dem anderen MAP-Decoder

als a-priori Information verwendet. Jeder einzelne Decoder erreicht zwar nur einen kleinen Gewinn, durch die iterative Verarbeitung besitzt das Gesamtsystem jedoch eine sehr hohe Leistungsfähigkeit. Zur symbolweisen MAP-Decodierung wird meist der BCJR-Algorithmus [Bahl u. a. 1974] eingesetzt, welcher zu den *message passing* Verfahren zählt (Übersicht siehe Anhang, Abb. A.2, S. 197).

2.2. Theoretische Grundlagen der Verteilten Quellencodierung

2.2.1. Klassifikation der Forschungsrichtungen

Die Verteilte Quellencodierung hat Bezug zu vielen verschiedenen Forschungsrichtungen (Abb. 2.8). Diese sollen hier kurz vorgestellt werden, um eine Begriffsklärung und Abgrenzung von benachbarten Forschungsgebieten zu ermöglichen.

konventionelle zentralisierte Quellencodierung — Anwendungsszenarien für die zentralisierte Quellencodierung beinhalten einen Sender (Encoder) sowie mehrere Empfänger (Decoder). Man spricht dabei auch von dem Rundfunkszenario (*broadcasting*). Dem zentralisierten Encoder stehen alle Informationen, beginnend von den Originaldaten bis hin zu Abhängigkeiten zu anderen Daten sowie dessen konkrete Realisierung, zur Verfügung.

Verteilte Quellencodierung (distributed source coding) — In Szenarien der Verteilten Quellencodierung gibt es keinen zentralisierten Encoder [Slepian u. Wolf 1973]. Mehrere Encoder bearbeiten unabhängig voneinander Daten unterschiedlicher miteinander korrelierter Quellen. Die Herausforderung besteht darin, die vorhandenen Ähnlichkeiten (Korrelation) am Decoder auszunutzen. Der Encoder besitzt dabei ausschließlich Kenntnis über die Statistik aller Quellen, jedoch nicht über deren konkrete Realisierungen.

Verteilte Quellencodierung mit Seiteninformation / asymmetrische Codierung — Bei der asymmetrischen Verteilten Quellencodierung handelt es sich um einen Sonderfall der Verteilten Quellencodierung, wobei die Realisierung einer Quelle direkt bzw. verlustlos codiert am Decoder zur Verfügung steht. Darauf aufbauend hat sich der Bereich der verlustbehafteten Verteilten Codierung mit

Seiteninformation gebildet, welcher die Grundlage der Verteilten Videocodierung darstellt [Slepian u. Wolf 1973][Wyner u. Ziv 1976].

Interaktive Codierung (interactive encoding) — Die Interaktive Codierung stellt eine Erweiterung der Verteilten Quellencodierung dar [Yang u. He 2008, 2010] [Meng u. a. 2011]. Es wird anstatt der unidirektionalen Kommunikation eine bidirektionale Kommunikation zwischen Encoder und Decoder verwendet, um die Rekonstruktion mit möglichst geringer Kommunikation zwischen Sender und Empfänger zu ermöglichen.

Netzwerk-Quellencodierung (network source coding) — Die Netzwerk-Quellencodierung bezieht die Netzwerkstruktur zwischen den multiplen Sendern und Empfängern mit in das Codierungsproblem ein [Fleming u. a. 2004][Ramamoorthy u. a. 2006]. Dabei können Übertragungsstationen nicht nur Nachrichten weiterleiten, sondern diese auch neu organisieren oder umcodieren. In der Verteilten Codierung wird im Gegensatz dazu von einer Punkt zu Punkt Verbindung ausgegangen.

gemeinsame Kanal- und Quellencodierung (joint channel source coding) — Bei der gemeinsamen Kanal- und Quellencodierung werden die Kanal- und Quellencodierung nicht wie üblich voneinander getrennt. Da die Verteilte Quellencodierung auch als Kanalcodierungsproblem aufgefasst werden kann, besteht hier eine enge Verbindung.

compressive sensing — *Compressive sensing* beschreibt die Beobachtung von Signalen \mathbf{x} , welche aus wenigen Basisfunktionen \mathbf{V} bestehen [Baraniuk 2007] [Candes u. Wakin 2008][Goyal u. a. 2008]. Es existiert somit eine dünn besetzte Repräsentation \mathbf{u} des Signals im Raum der Basisfunktionen. Ziel ist es, das Rekonstruktionssignal $\hat{\mathbf{x}}$ zu bestimmen, wobei ausschließlich die Beobachtungen \mathbf{y} und Basisfunktionen bekannt sind. Die Beobachtungen bilden Linearkombinationen Φ der Elemente des Originalsignals \mathbf{x} . Dieser Ansatz kann auch zur Datenkompression eingesetzt werden, wobei der Encoder Beobachtungen durch einfache Linearkombinationen erzeugt. Der Decoder hat dabei ein komplexes Optimierungsproblem zu lösen, um das Originalsignal zu rekonstruieren. In diesem System muss ausschließlich der Decoder Kenntnis über die Basisfunktionen Φ besitzen, welche man auch als Seiteninformation ansehen kann.

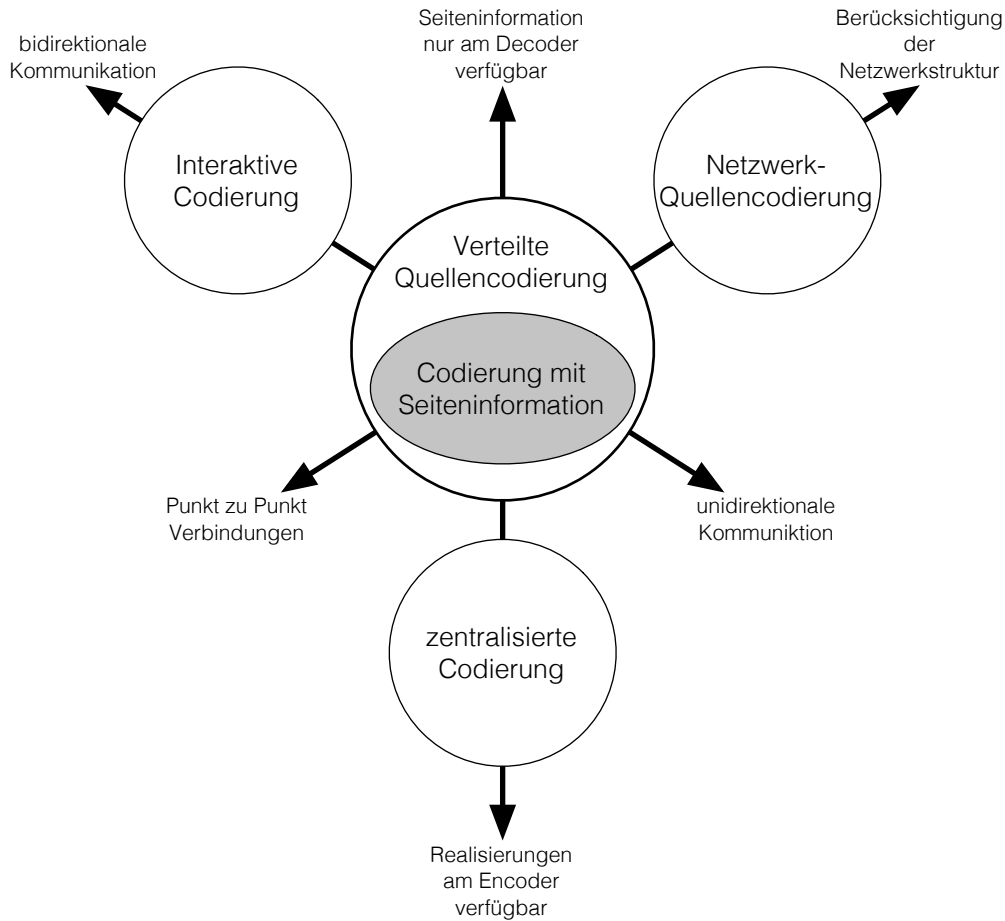


Abbildung 2.8.: Beziehung des Forschungsgebietes der Verteilten Quellencodierung zu verwandten Forschungsgebieten

2.2.2. Verteilte Quellencodierung

Die Grundlage für die verlustlose Verteilte Quellencodierung wurde durch David Slepian und Jack K. Wolf [Slepian u. Wolf 1973] gelegt. Sie analysierten die erreichbaren Datenraten (R_X, R_Y) für Systeme mit getrennten und verbundenen Encodern und Decodern (Abb. 2.9). Der besonders interessante Fall der getrennten Encodierung und gemeinsamen Decodierung entspricht dem Prinzip der Verteilten Codierung. Es konnte gezeigt werden, dass die Summendatenrate $R_X + R_Y$ gleich der Verbundentropie ist (Gl. 2.18). Dies entspricht der klassischen gemeinsamen Encodierung, wobei jedoch die Encoder getrennt sind. Als weitere Bedingung muss, im Gegensatz zur zentralisierten Quellencodierung, jede Datenrate R_X, R_Y größer als die bedingte Entropie sein (Gl. 2.17).

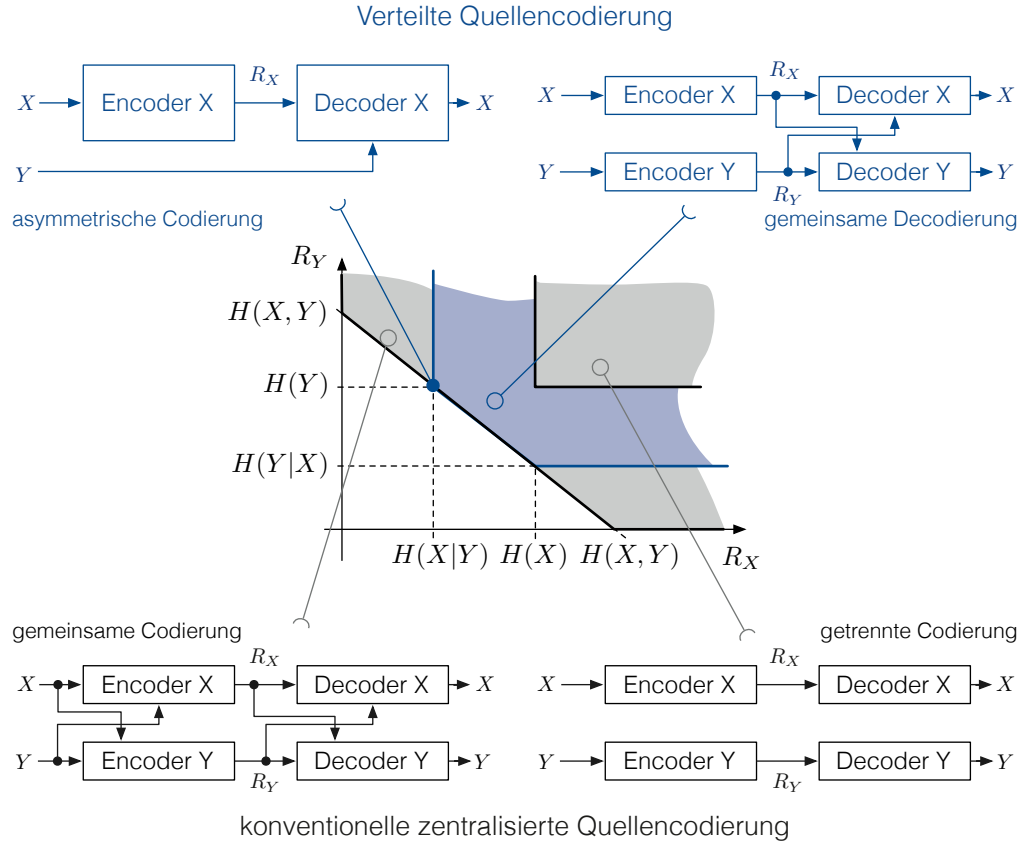


Abbildung 2.9.: Erreichbare Ratenregionen für verschiedene Szenarien der gemeinsamen und getrennten Codierung bzw. Decodierung

$$R_X \geq H(X|Y); \quad R_Y \geq H(Y|X) \quad (2.17)$$

$$R_X + R_Y \geq H(X, Y) \quad (2.18)$$

Einen für die Verteilte Videocodierung wichtigen Sonderfall nimmt die asymmetrische Codierung ein. Die Symbole der Quelle Y stehen am Decoder zur Verfügung bzw. wurden unabhängig von X codiert ($R_Y \geq H(Y)$). Aaron D. Wyner und Jacob Ziv [Wyner u. Ziv 1976] betrachteten die verlustbehaftete Quellencodierung für diesen asymmetrischen Systemaufbau (Abb. 2.10). Sie analysierten die Rate-Distortion Funktion für die Fälle, dass die Seiteninformation Y am En- und Decoder $R_{X|Y}(D)$ oder nur am Decoder $R^*(D)$ vorlag. Es konnte gezeigt werden, dass für Gauß-verteilte kontinuierliche Zufallsvariablen die Datenrate, unabhängig von der Verfügbarkeit der Seiteninformation am Decoder, gleich ist (Gl. 2.19). Dies ist von

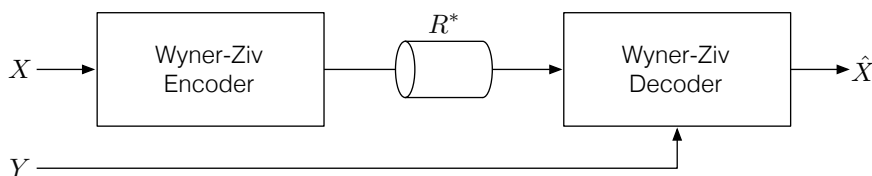


Abbildung 2.10.: Wyner-Ziv Codierung, Seiteninformation Y ist ausschließlich am Decoder verfügbar

fundamentaler Bedeutung für die Verteilte Videocodierung.

$$R^*(D) = R_{X|Y}(D) \quad (2.19)$$

2.3. Anwendungsszenarien für die Verteilte Videocodierung

Die Anwendungsszenarien der Verteilten Videocodierung sind sehr vielfältig, einen ausführlichen Überblick liefern [Pereira u. a. 2006, 2008] und [Dufaux u. a. 2009], wobei hier die wichtigsten Anwendungsgebiete kurz klassifiziert werden sollen (Abb. 2.11).

Die Verteilte Videocodierung hat drei wesentliche Eigenschaften, welche den meisten Anwendungsszenarien zugrunde liegen. Zum einen bietet sie die Möglichkeit, Systeme mit geringer Encoderkomplexität zu entwerfen. Darüber hinaus ist es möglich, Encoder getrennt voneinander zu betreiben und zudem zeigt die Verteilte Videocodierung keine Drift. Von Drift spricht man, wenn sich ein Fehler in einem Bild auf weitere Bilder fortsetzt, auch wenn die Decodierung fehlerfrei durchgeführt werden konnte.

Videocodierung auf ressourcenarmen Geräten (low complexity video coding) — Die Videocodierung auf ressourcenarmen Endgeräten wird durch eine getrennte Encodierung aufeinanderfolgender Frames einer Videosequenz, aber gemeinsamer Decodierung, erreicht [Girod u. a. 2005][Artigas u. a. 2007a]. Die Encoderkomplexität wird stark reduziert, da keine Bewegungsschätzung am Encoder notwendig ist. Ein populäres Gerät ist das kabellose Endoskop, welches

über eine Funkschnittstelle die codierte Sequenz an einen Empfänger außerhalb des Patienten überträgt [Deligiannis u. a. 2011].

Videocodierung mehrerer Ansichten (multiview video coding) — Die Codierung mehrerer Ansichten einer Szene ermöglicht auch die Auswertung der Abhängigkeiten zwischen den Sichten. Wenn dies aufgrund der räumlichen Trennung oder beschränkter Ressourcen der Encoder nicht möglich ist, bildet die Verteilte Videocodierung einen vielversprechenden Lösungsansatz [Chen u. a. 2008][Varodayan u. a. 2007].

Videosensornetzwerke (video sensor networks) — Die Beobachtung von Wildbewegungen in einem Waldgebiet kann durch Videosensornetzwerke erfolgen. Kleine sehr ressourcenarme Sensorknoten nehmen Bildsequenzen auf und übertragen diese per Funk zu einer zentralen Station [Puri u. a. 2006]. Hierbei spielt die geringe Encoderkomplexität, die Berücksichtigung der Ähnlichkeit zwischen den Sichten der Sensorknoten und die fehlerrobuste Übertragung, welche durch die Verteilte Videocodierung bereitgestellt wird, eine wichtige Rolle.

Repräsentationswechsel (bit stream switching) — Der Wechsel des Empfängers von einem codierten Videodatenstrom (z.B. H.264) zu einem anderen ist nur bei bestimmten Frametypen möglich (I,SP,SI). Mit Hilfe von Wyner-Ziv Frames (WZ) lässt sich dieses Problem mit einer geringeren Anzahl WZ-Frames im Vergleich zu SP-Frames lösen [Sun u. a. 2006][Guo u. a. 2006a].

skalierbare Videocodierung (scalable coding) — Die Verteilte Videocodierung kann zur skalierbaren Videocodierung bezüglich der Qualität sowie zeitlicher und örtlicher Auflösung verwendet werden. Der konkrete Basiscodec ist nicht ausschlaggebend, dieser muss ausschließlich eine ausreichende Ähnlichkeit zu der zu erreichenden Qualität bzw. Auflösung sicherstellen [Wang u. a. 2004, 2006][Wang u. Ortega 2004].

Qualitätsverbesserung (enhancement coding) — Die Qualität von analogen Videoübertragungen lässt sich durch die zusätzliche Übertragung eines Korrekturdatenstromes, nach den Prinzipien der Verteilten Videocodierung, verbessern. Dies sichert die Abwärtskompatibilität und ist der Qualitätsskalierbarkeit sehr ähnlich [Girod u. a. 2005][Rane u. a. 2004, 2005].

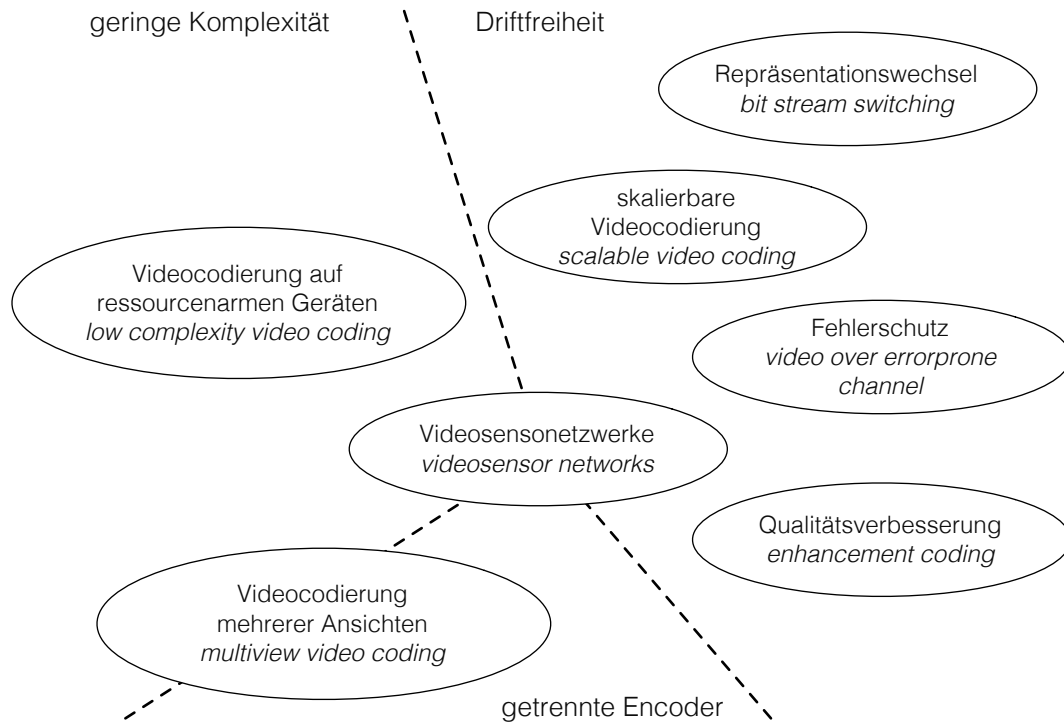


Abbildung 2.11.: Anwendungsszenarien, gruppiert nach der zugrunde liegenden Eigenschaft der Verteilten Videocodierung

Fehlerschutz (video over errorprone channel) — Die Verteilte Videocodierung gilt als sehr robust gegenüber Übertragungsfehlern und kann zum Schutz von konventionellen Videoübertragungen (MPEG.2, MPEG.4, H.264) eingesetzt werden [Aaron u. a. 2003b].

Die Verteilte Videocodierung findet, wie gezeigt wurde, viele Anwendungsgebiete. Die vorliegende Arbeit konzentriert sich auf Systeme mit geringer Encoderkomplexität. Nachfolgend werden ausgewählte Systeme exemplarisch vorgestellt.

2.4. Verteilte Videocodierungssysteme mit geringer Komplexität

In den letzten zehn Jahren gewann die Verteilte Videocodierung erhöhtes Forschungsinteresse, da die Umsetzung durch neu entwickelte leistungsfähige Codierungsverfahren möglich wurde. In dieser Zeit wurden mehrere Systeme zur Verteil-

ten Videocodierung mit geringer Encoderkomplexität entwickelt, welche im Folgenden kurz beschrieben werden. Anschließend werden in den Abschnitten 2.4.2 und 2.4.3 die Entwicklungen im Bereich der Seiteninformationsgenerierung und Codierung vorgestellt.

2.4.1. Überblick: Verteilte Videocodierungssysteme mit geringer Encoderkomplexität

PRISM — Das Codierungssystem PRISM ist eines der ersten Systeme zur Verteilten Quellencodierung. Hierbei wird die verlustbehaftete Codierung mit Seiteninformation (Wyner-Ziv Codierung) durch Quantisierung und nachfolgende Slepian-Wolf Codierung (SW-Codierung) erreicht. Die Quantisierungssymbole werden zur Codierung in mehreren Gruppen (*cosets*) eingeteilt, wobei die Symbole einer Gruppe einen möglichst großen Abstand zueinander besitzen (Abb. 2.12). Der Gruppenindex wird übertragen. Am Decoder wird das Symbol aus der Gruppe mit dem geringsten Abstand zur Seiteninformation als Rekonstruktionswert gewählt. Bei der Verwendung mehrerer Seiteninformationen wird darüber hinaus anhand eines Codes zur Fehlererkennung (*hash*) die passende Seiteninformation ausgewählt [Puri u. Ramchandran 2002, 2003][Puri u. a. 2006, 2007].

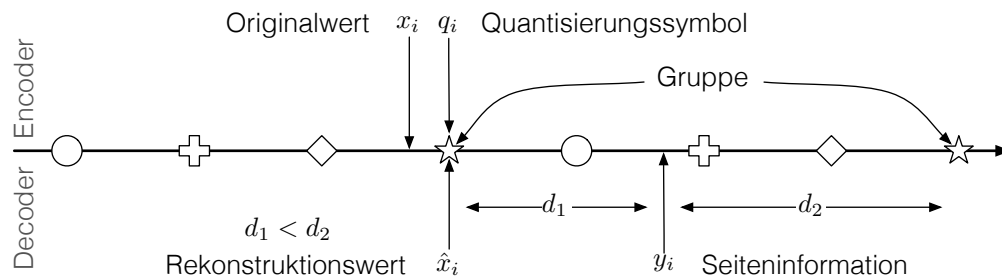


Abbildung 2.12.: Codierung in Untergruppen (*cosets*) und Rekonstruktion in PRISM

Stanford — Die in Stanford entwickelte Architektur (Abb. 2.13) ist sehr weit verbreitet. Es werden zwei Frametypen, die Key- und Wyner-Ziv Frames, eingeführt, wobei die Key-Frames mit konventionellen Verfahren codiert werden. Auch in dieser Architektur wird die WZ-Codierung durch eine Quantisierung

und nachfolgender SW-Codierung umgesetzt. Die SW-Codierung basiert auf Kanalcodes, wie Turbocodes oder LDPCA Codes (*low density parity check accumulate code*), wobei die Coderate über einen Rückkanal angepasst wird. Die Seiteninformation wird, basierend auf den decodierten Key- oder WZ-Frames, durch temporale Interpolation oder Extrapolation gewonnen [Girod u. a. 2005] [Aaron u. a. 2002, 2003a, 2004a, 2006] [Aaron u. Girod 2004]. Die örtliche Korrelation kann optional durch eine geeignete Transformation ausgenutzt werden.

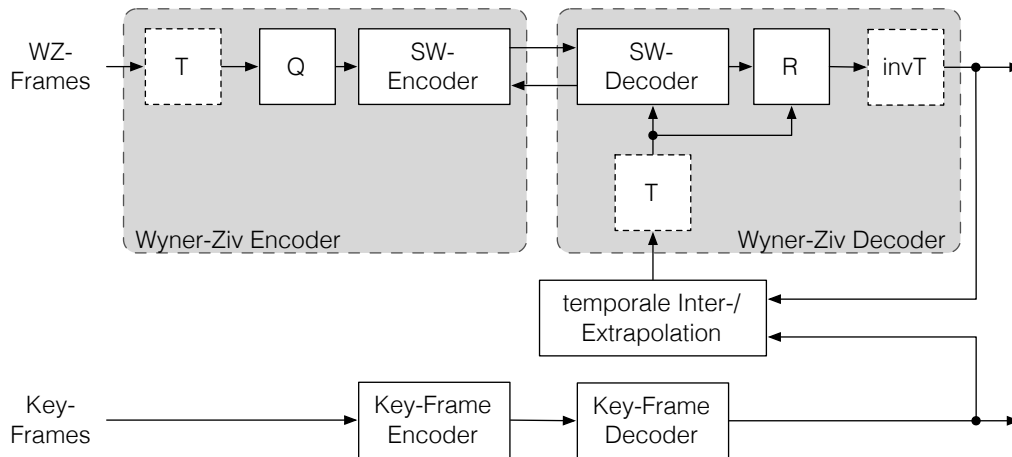


Abbildung 2.13.: Stanford-Architektur (Q - Quantisierung, R - Rekonstruktion, T/invT - Hin-/Rücktransformation)

DISCOVER — Der Discover Codec entspricht in der Grundstruktur der Stanford-Architektur [Artigas u. a. 2007a][DISCOVER 2007]. Der Codec wurde zu Beginn als pixelbasiertes System umgesetzt [Ascenso u. a. 2005]. Später wurde zur Steigerung der Leistungsfähigkeit die örtliche Dekorrelation mittels diskreter Kosinustransformation (DCT) verwendet [Brites u. a. 2006a]. Viele der aktuell entwickelten Verfahren beziehen sich auf den DISCOVER Codec.

VIPER — Ein Codec, welcher intensiv Gebrauch von dem Rückkanal macht (*Backward channel aware coding*), wurde vom VIPER-Lab vorgestellt [Liu u. a. 2006, 2007]. In diesem System werden Bewegungsvektoren vom Decoder zum Encoder übertragen, um eine bewegungskompensierte prädiktive Codierung zu ermöglichen, was zu einer Steigerung der Leistungsfähigkeit führt. Dieses System bewegt sich zwischen den Forschungsrichtungen der „Verteilten Quellencodierung“ und der „Interaktiven Quellencodierung“.

Codecs auf Basis von Compressive Sensing — Das Forschungsgebiet *Compressive sensing* (CS) findet zunehmend Beachtung. Basierend auf CS-Verfahren wurden Videocodecs mit geringer Encoderkomplexität entwickelt [Prades-Nebot u. a. 2009][Do u. a. 2009][Chen u. a. 2010]. Die zur Rekonstruktion notwendigen Basisfunktionen werden aus den benachbarten Key-Frames oder der Seiteninformation gewonnen. Diese Systeme zeigten eine etwas geringere Leistungsfähigkeit als der DISCOVER Codec.

In allen hier kurz vorgestellten Systemen gibt es zwei wichtige Komponenten, die Generierung der Seiteninformation und die Codierung. Bekannte Verfahren aus diesen beiden Gruppen werden in den nachfolgenden Abschnitten 2.4.2 und 2.4.3 beschrieben.

2.4.2. Verfahren zur Generierung der Seiteninformation

Die Art der Generierung der Seiteninformation hängt stark von dem Anwendungsszenario ab. Hier erfolgt eine Beschränkung auf Systeme mit geringer Encoderkomplexität. Für diese Systeme bietet sich, aufgrund der hohen temporalen Korrelation, die zeitliche Inter- bzw. Extrapolation an. Weiterhin finden sich in der Literatur auch Ansätze, welche die örtliche Korrelation berücksichtigen.

Zeitliche Interpolation

Für die zeitliche Interpolation ist es notwendig, die Bewegung zu berücksichtigen (Anhang A.2.1, S. 198). Dies kann durch ein Blockmodell erfolgen, indem die Bewegungsvektoren mittels Blockmatching bestimmt werden (BiMESS [Ascenso u. a. 2005]). In Weiterentwicklungen wurden eine hierarchische Bewegungsschätzung mit unterschiedlichen Blockgrößen [Ascenso u. Pereira 2008] oder auch die Bestimmung eines sehr dichten Bewegungsvektorfeldes [Sofke u. a. 2009], zur Steigerung der Seiteninformationsqualität, eingeführt. Diese Verfahren setzen die Linearität der Bewegung voraus. Diese Voraussetzung konnte in [Petrazzuoli u. a. 2010a, b] durch die Verwendung eines zeitlichen Bewegungsmodells höherer Ordnung (Polynom 3. Grades) aufgebrochen werden.

Neben den blockbasierten Bewegungsmodellen werden in der Literatur auch netzbasierte Modelle [Kubasov u. Guillemot 2006], 3D-Modelle für statische Szenen



Abbildung 2.14.: Prinzip der Generierung der Seiteninformation, basierend auf der temporalen Korrelation

[Maitre u. a. 2006, 2007] oder auch Gesichtsmodelle, angepasst an Videokonferenzanwendungen [Artigas u. Torres 2005b, 2006], beschrieben.

Da am Decoder neben den zuvor decodierten Frames auch decodierte Anteile des aktuellen Frames zur Verfügung stehen, können diese in einem iterativen bewegungskompensierten Verfeinerungsprozess (*refinement*) genutzt werden (Abb. 2.15, [Artigas u. Torres 2005a][Adikari u. a. 2006][Martins u. a. 2009][Tomé u. Pereira 2011]). Dies stellt eine Erweiterung der Stanford-Architektur dar.

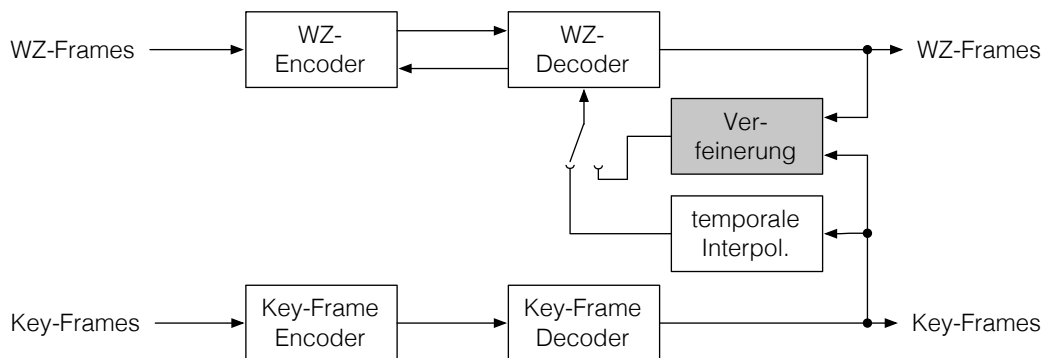


Abbildung 2.15.: Verfeinerung der Seiteninformation nach erfolgreicher Decodierung (MCR, *motion compensated refinement*)

Zeitliche Extrapolation

Die zeitliche Extrapolation bietet im Vergleich zur Interpolation den Vorteil, dass keine strukturbedingte Verzögerung in das Codierungssystem eingefügt wird, welche durch das Warten auf nachfolgende decodierte Frames entsteht. Die Extrapolation zeigt jedoch eine schlechtere Seiteninformationsqualität als die zeitliche Interpolation [Natário u. a. 2005][Borchert u. a. 2007, 2008][Borchert 2010].

Örtliche Interpolation

Neben der hohen zeitlichen Korrelation in einer Videosequenz wird auch die Ähnlichkeit benachbarter Bildpunkte genutzt, um die Seiteninformation zu bestimmen [Tagliasacchi u. a. 2006][Ye u. a. 2009]. Da die örtliche Korrelation jedoch oft wesentlich geringer als die zeitliche Korrelation ist, werden diese Verfahren meist mit einer temporalen Interpolation kombiniert.

Hashbasierte Ansätze

Eine Erweiterung der Stanford-Architektur stellen die hashbasierten Ansätze dar (Abb. 2.16, [Aaron u. a. 2004b]). Die Grundidee besteht darin, einen Teil des WZ-Frames mit konventionellen Codierungsverfahren zu übertragen. Dieser Anteil kann am Decoder zur Unterstützung der Generierung der Seiteninformation verwendet werden oder auch Teile des rekonstruierten Bildes ersetzen. Dies führt zu einer Steigerung der Leistungsfähigkeit besonders für Sequenzen mit einer geringen temporalen Korrelation. Bei dem Hash kann es sich um einzelne Pixel [Park u. a. 2012], hochfrequente DCT-Koeffizienten [Aaron u. Girod 2004][Ascenso u. Pereira 2007] oder einzelne Blöcke im Bild [Trapanese u. a. 2005] handeln.

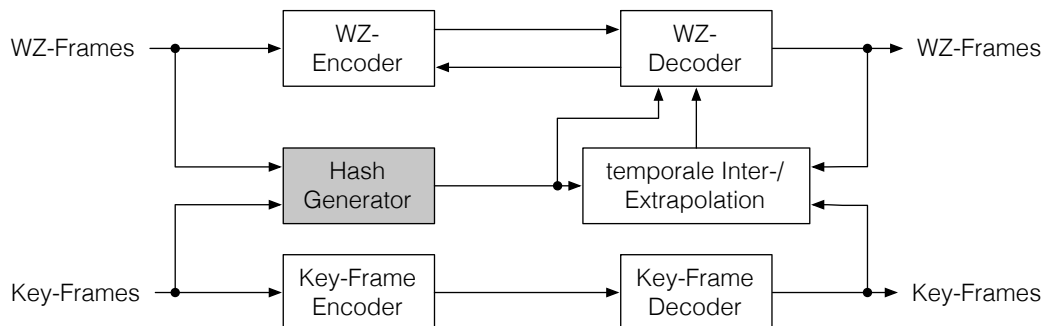


Abbildung 2.16.: Stanford-Architektur mit Übertragung eines Hash zur Unterstützung der Bewegungsschätzung sowie der Decodierung

Kombination mehrerer Seiteninformationen

Stehen am Decoder mehrere Seiteninformationsgenerierungsverfahren zur Verfügung, so kann die Leistungsfähigkeit durch deren Kombination verbessert werden

[Kubasov u. a. 2007b][Huang u. a. 2009]. Die blockweise Auswahl der Seiteninformation mit der höchsten Korrelation wird in [Varodayan u. Girod 2010], basierend auf *doping*, beschrieben. Unter *doping* versteht man die fehlerfreie Übertragung eines Teils der Originalinformation.

2.4.3. Wyner-Ziv und Slepian-Wolf Codierung

Die verlustbehaftete Verteilte Quellencodierung mit Seiteninformation wird in vielen bekannten Systemen durch eine Quantisierung und nachfolgender SW-Codierung realisiert.

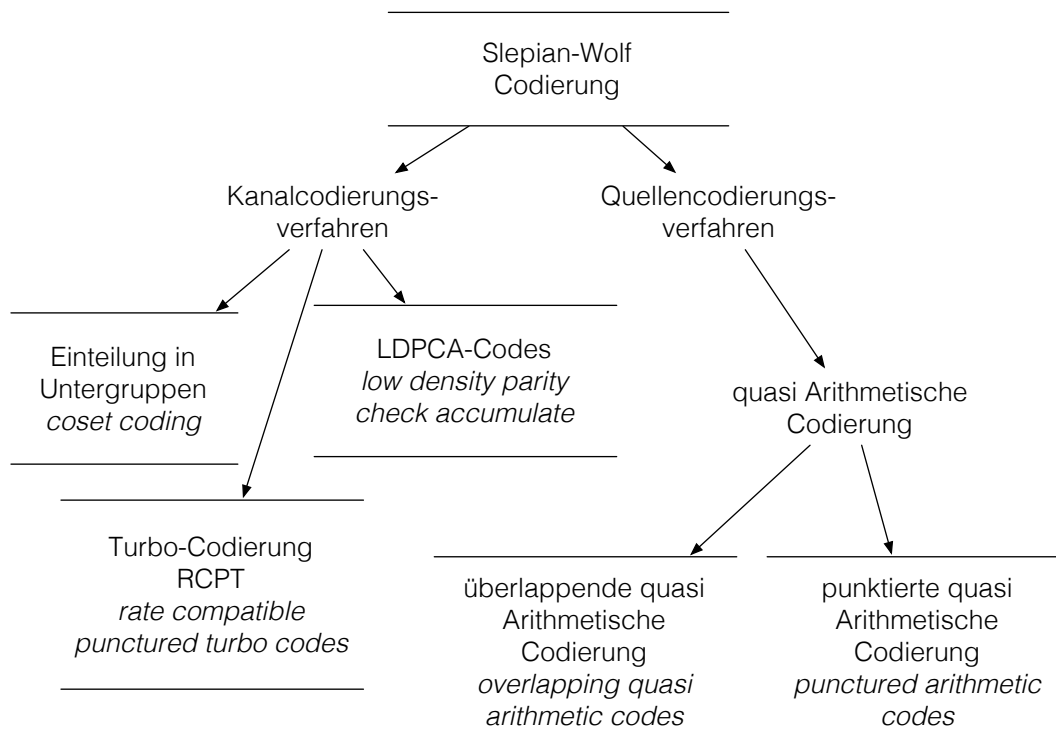


Abbildung 2.17.: Klassifikation der Slepian-Wolf Codierungsverfahren

Eines der ersten Systeme zur Verteilten Videocodierung (PRISM, [Puri u. a. 2006], [Puri u. a. 2007] [Pradhan u. Ramchandran 1999]) setzte die SW-Codierung durch die Einordnung der Symbole in Gruppen um (*coset coding*, Abb. 2.12). Dabei wird nur der Gruppenindex übertragen. Diese Herangehensweise kann auch als Kanalcodierung betrachtet werden, womit in weiteren Systemen das Slepian-Wolf Quellen-

codierungsproblem als Kanalcodierungsproblem formuliert und auch gelöst wurde (Abb. 2.17).

Erste Ansätze nutzen die sehr leistungsfähige Turbo-Codierung [Garcia-Frias u. Zhao 2001][Aaron u. Girod 2002]. Die Herausforderung hierbei besteht in der Anpassung der Coderate R_c beziehungsweise Quellenrate R_s an die Korrelation zwischen den Originalsymbolen und den Symbolen der Seiteninformation. Ist die Korrelation gering, so ist eine leistungsfähigere Codierung mit geringerer Coderate R_c notwendig. Die Quellenrate R_s und auch die Menge der übertragenen Daten ist somit größer.

Die Anpassung der Leistungsfähigkeit der Turbo-Codierung an die Qualität der Seiteninformation wurde durch die Verwendung von ratenadaptierbarer Turbocodes (RCPT, *rate compatible punctured turbo codes*, [Rowitch u. Milstein 2000]) erreicht. Diese basieren auf den anpassbaren Faltungscodes (RCPC, *rate compatible punctured convolutional codes*, [Hagenauer 1988]). Bei dieser Art der Codierung ist der Code einer höheren Coderate (kleinere Quellenrate) immer Teil des Codes kleinerer Coderate (höhere Quellenrate). So kann inkrementell die Leistungsfähigkeit durch Übertragung weiterer Korrektursymbole $\mathbf{c}^{*(i)}$ erhöht werden (Abb. 2.18). Der Decoder kann aus den bekannten Korrektursymbolen $\mathbf{c}^{(i-1)}$ und aus den neu empfangenen Daten $\mathbf{c}^{*(i)}$ die Prüfsymbole für einen leistungsfähigeren Code $\mathbf{c}^{(i)}$ zusammensetzen, um eine erfolgreiche Decodierung zu erreichen (Abb. 2.18).

Weiterhin wurden die leistungsfähigen LDPC Codes (*low density parity check*) modifiziert, um auch diese in der Coderate anpassen zu können und somit für die Verteilte Quellencodierung nutzbar zu machen (LDPCA, *low density parity check accumulate*, [Varodayan u. a. 2005, 2006]). In [Westerlaken u. a. 2007] konnte gezeigt werden, dass die LDPCA Codes für die Bitebenen- und die symbolweise Codierung die gleiche Leistungsfähigkeit zeigen, womit die Verwendung eines höheren Alphabets nicht notwendig ist.

Neben den bekannten Codierungsverfahren aus der Kanalcodierung wurden auch Verfahren aus der Quellencodierung herangezogen, um das Problem der Verteilten Quellencodierung zu lösen. Diese bieten im Allgemeinen den Vorteil, dass Quelleneigenschaften, wie zum Beispiel die Quellenentropie, berücksichtigt werden. Bekannte Verfahren aus diesem Bereich sind die überlappende quasi arithmetische Codierung (*overlapping quasi arithmetic codes*, [Artigas u. a. 2007b]) oder auch die punktierte quasi arithmetische Codierung [Malinowski u. a. 2009]. Die punktierte Codierung

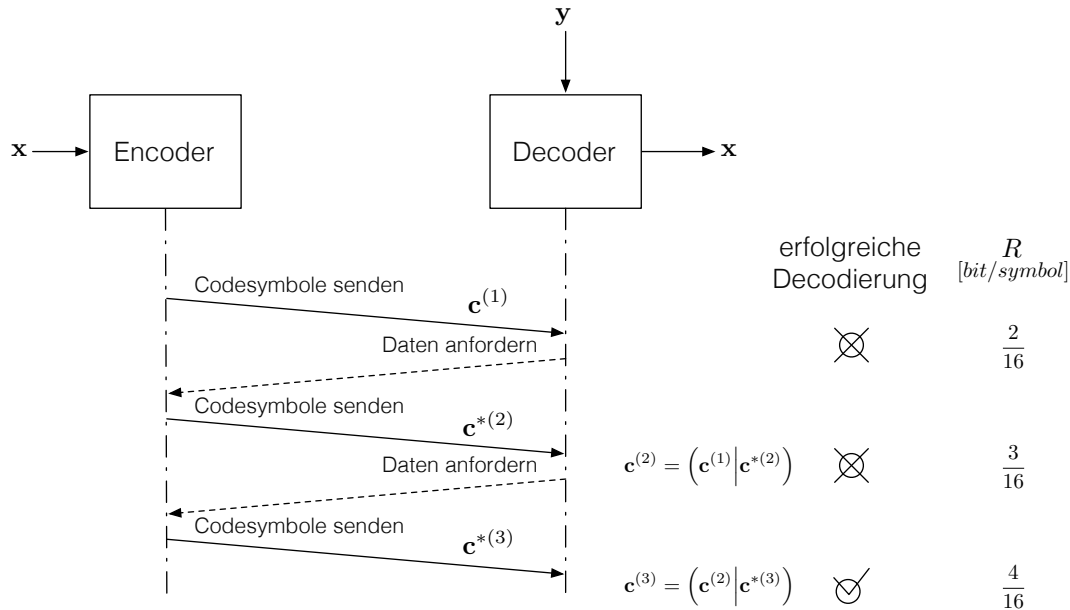


Abbildung 2.18.: Kommunikation zwischen Encoder und Decoder bei der Rückkanal-gestützten Slepian-Wolf Codierung

ist einfacher in der Leistungsfähigkeit anzupassen. Beide Verfahren zeigen eine gute Leistungsfähigkeit besonders bei kurzen Symbolfolgen.

In vielen Systemen zur Verteilten Videocodierung wird eine Transformation zur Berücksichtigung der örtlichen Korrelation eingesetzt (Abschn. 2.4.1). Neben der DCT (*discrete cosine transform* [Aaron u. a. 2004a]) wurden auch Untersuchungen zur Verwendung der Wavelet-Transformation (DWT, *discrete wavelet transform*) durchgeführt [Guo u. a. 2006b][Yang u. a. 2006]. Diese konnte sich in dem Forschungsgebiet nicht durchsetzen.

2.4.4. Modellierung

Die Leistungsfähigkeit der Wyner-Ziv Codierung ist stark von der Modellierung der Korrelation zwischen Seiteninformation und Originaldaten abhängig. Neben dem oft verwendeten Modell des Laplace-verteilten Rauschens (Gl. 2.20) wurden in [Westerlaken u. a. 2006] weitere Rauschverteilungen $p_N(n)$ untersucht. Eine wesentliche Verbesserung konnte in [Brites u. a. 2006b] durch die Bestimmung des

Modellparameters λ der Laplaceverteilung auf Frame-, Block- und Pixelebene erreicht werden. Damit wird die Stationarität des Rauschens nicht mehr vorausgesetzt.

$$p_N(n) = \frac{1}{2} \lambda e^{-\lambda|n|} \quad (2.20)$$

Die verfeinerte Schätzung des Modellparameters während der Decodierung wurde darüber hinaus in [Luong u. a. 2011] für ein iteratives Bitebenen übergreifendes Decodierungsverfahren betrachtet, womit leichte Steigerungen der Leistungsfähigkeit erreicht wurden.

2.4.5. Zusammenfassung

In diesem Abschnitt 2.4 wurde ein Überblick über Verteilte Videocodierungssysteme mit geringer Encoderkomplexität und den entsprechenden Verfahren gegeben. Die stetige Weiterentwicklung hat zu einer Steigerung der Leistungsfähigkeit, im Wesentlichen durch die Verbesserung der Seiteninformationsgenerierung und -verfeinerung und durch Verwendung einer hash- und transformationsbasierten Codierung, geführt. Die Encoderkomplexität wird durch Anwendung dieser Verfahren zum Teil erhöht.

In den folgenden Kapiteln 3, 4 und 5 wird ein Codierungssystem mit geringer Encoderkomplexität, auch für komplexe Bewegungsstrukturen, neu entwickelt und anhand von Zwischenergebnissen evaluiert. Die Leistungsfähigkeit des Gesamtsystems wird abschließend in Kapitel 6 untersucht.

Durch den Rückkanal werden DVC-Systeme unpraktikabel für viele Anwendungsszenarien. Viele Ansätze versuchen, dieses Problem im Rahmen der Ratenkontrolle zu lösen. In Kapitel 7 werden darüber hinaus Möglichkeiten zur Verbesserung der Flexibilität des Decoders vorgeschlagen, um die Verwendung des Rückkanals zu vermeiden.

3

Kapitel 3

Rostock Distributed Video Codec – *ρDeViCe*

Erweiterungen des Codierungssystems, wie die Einführung der diskreten Cosinustransformation (DCT), werden in diesem Kapitel hinsichtlich der Auswirkungen auf die Encoderkomplexität beurteilt. Basierend darauf werden die Struktur des in dieser Arbeit neu entwickelten Rostock Distributed Video Codec – *ρDeViCe* beschrieben und grundlegende Entscheidungen für die wesentlichen Komponenten getroffen.

Die sehr geringe Encoderkomplexität stellt eine wesentliche Randbedingung dar.

3.1. Anforderungen und Zielsetzungen

Die Verteilte Videocodierung ist Anfang dieses Jahrtausends stark in den Fokus der Forschungsgruppen getreten. Dabei bestand vielfach die Zielsetzung, ein Codierungssystem mit geringer Encoderkomplexität zu entwickeln (Abschn. 2.3). Bis heute sind im Wesentlichen zwei große Probleme ungelöst. Dies ist zum einen die geringe RD-Performanz (*rate distortion performance*) im Vergleich zur konventionellen Videocodierung und zum anderen der Zwang zur Nutzung eines Rückkanals zur genauen Ratenkontrolle.

Zur Lösung dieser Probleme soll mit der Entwicklung von *ρDeViCe* ein Beitrag geliefert werden, wobei folgende Anforderungen formuliert werden:

Encoderkomplexität — Die Komplexität des Encoders soll auf einem möglichst niedrigen Niveau belassen werden, um sehr ressourcenschwache Endgeräte zu berücksichtigen.

RD-Performanz — Eine hohe RD-Performanz soll auch für Sequenzen mit komplexen Bewegungsstrukturen angestrebt werden. Die RD-Performanz soll, im Hinblick auf die Komplexität des Encoders, mit den konventionellen Videocodierungsverfahren vergleichbar sein.

Flexibilität — Die Freiheitsgrade des Decoders und somit dessen Flexibilität in Bezug auf die Art der Decodierung sollen gesteigert werden. Neben den bekannten Decodierungsverfahren sollen weitere Ansätze entwickelt werden, welche den Schritt zu rückkanalfreien Systemen erleichtern.

Unter Berücksichtigung dieser Zielsetzungen wird in den folgenden Abschnitten die Systemstruktur (Abschn. 3.3), mit besonderer Rücksicht auf die Verwendung einer örtlichen Transformation (Abschn. 3.2), entwickelt.

3.2. Einfluss der DCT auf die Komplexität eines DVC-Encoders

Konventionelle Bild- und Videokompressionssysteme nutzen die diskrete Cosinus-Transformation Typ-II (DCT-II), um die örtlichen Korrelationen auszuwerten und somit die Leistungsfähigkeit des Gesamtsystems zu steigern. Die DCT-II wird auch in Systemen zur Verteilten Videocodierung eingesetzt [Artigas u. a. 2007a].

Unter der Zielsetzung einer geringen Encoderkomplexität soll nachfolgend der Einfluss der DCT-II auf die Komplexität eines Wyner-Ziv (WZ) Encoders diskutiert werden.

3.2.1. Komplexität der DCT

In der Bilddatenkompression spielt die zweidimensionale DCT-II eine große Rolle [Ohm 2004, S.118][Richardson 2003, S.45][Strutz 2002, S.90], welche direkt durch Anwendung von Gleichung 3.2 bestimmt werden kann.

Da diese direkte Umsetzung der DCT-II sehr rechenaufwendig ist, kann die orthogonale Separierbarkeit der 2D-DCT-II genutzt werden, um diese in zwei 1D-DCT-II zu zerlegen (Gl. 3.1, 3.3, *row-column method*, \mathbf{X} Matrix mit Elementen $x_{DCT,k,l}$).

$$\text{eindimensionale DCT: } \mathbf{x}_{DCT} = \mathbf{x} \cdot \mathbf{M} \quad (3.1)$$

$$\begin{aligned} \text{zweidimensionale DCT: } x_{DCT,k,l} &= \sum_i \sum_j x_{i,j} b_{i,j}^{k,l} \\ &= \mathbf{X} \bullet \mathbf{B}^{k,l} \end{aligned} \quad (3.2)$$

$$\text{zweidimensionale separierte DCT: } \mathbf{X}_{DCT} = \mathbf{M}^T \cdot \mathbf{X} \cdot \mathbf{M} \quad (3.3)$$

Dabei ist der Zeilenvektor \mathbf{x} bzw. die Matrix \mathbf{X} das ein-/zweidimensionale Originalsignal, jede Spalte von \mathbf{M} enthält eine eindimensionale Basisfunktion und $\mathbf{B}^{k,l}$ stellt eine zweidimensionale Basisfunktion für den Transformationskoeffizienten $x_{DCT,k,l}$ dar.

Neben den zuvor beschriebenen direkten und separierten Anwendungen der DCT-II gibt es auch eine Reihe optimierter Verfahren zur Berechnung der DCT. Eine Übersicht beschleunigter Verfahren ist in Tabelle 3.1 für die in der Bilddatenkompression gängigen Blockgrößen von $4 \times 4 \text{ px}^2$ bzw. $8 \times 8 \text{ px}^2$ zusammengefasst [Richter 2006]. Je kleiner die Blockgröße ist, desto geringer ist die Anzahl der durchzuführenden Additionen und Multiplikationen pro Pixel. Eine sehr schnelle modifizierte DCT wird in [Malvar u. a. 2003] beschrieben, welche auch in dem Videocodierungsstandard H.264/AVC [Wiegand u. a. 2003] Anwendung findet. Hierbei handelt es sich um eine Festkommaumsetzung, welche bei einer Blockgröße von $4 \times 4 \text{ px}^2$ nur eine binäre Schiebeoperation und 4 Additionen pro Pixel benötigt.

3.2.2. Komplexität eines DVC Encoders

Der Einfluss der DCT auf die Komplexität des Encoders kann nur durch Betrachtung des gesamten Encoders ermittelt werden. Die Abbildung 3.1 zeigt die Struktur eines WZ-Encoders mit bzw. ohne DCT. Im Bereich der DCT basierten DVC Codecs wird meist eine Blockgröße von $4 \times 4 \text{ px}^2$ bevorzugt [Artigas u. a. 2007a].

Blockgröße Operation je Pixel	$4 \times 4 \text{ px}^2$		$8 \times 8 \text{ px}^2$	
	×	+	×	+
direkt (Gl. 3.2)	16	15	64	63
separiert (Gl. 3.3)	8	6	16	14
[Lee 1984]	2	4.5	3	7.25
[Loeffler u. a. 1989]			2.75	7.25
[Feig u. Winograd 1992]			0.8125	7.21
[Lee u. Huang 1994]	1.25	4.5	2	7.25
[Malvar u. a. 2003]	1 (shift)	4		

Tabelle 3.1.: Anzahl der arithmetischen Operationen zur Durchführung der DCT-II je Pixel für die Blockgrößen $4 \times 4 \text{ px}^2$ und $8 \times 8 \text{ px}^2$

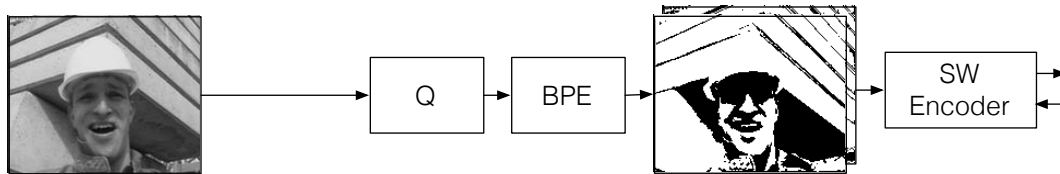
Wird die DCT verwendet, so müssen die zusätzliche Komplexität der Transformation und der Einfluss auf die Komplexität der nachgelagerten Verarbeitungsstufen des Encoders berücksichtigt werden (Abb. 3.1).

Quantisierung (Q) — Die Komplexität der Quantisierung kann durch den Einsatz der DCT verringert werden, da bei geringer Datenrate nicht alle DCT-Bänder übertragen und somit nicht quantisiert werden müssen. In Abbildung 3.1 ist dies beispielhaft für den RD-Punkt $Q_i = 4$ des DISCOVER Codecs dargestellt [Brites u. a. 2006a][DISCOVER 2007]. Hierbei reduziert sich die Anzahl der zu quantisierenden Werte auf $\frac{10}{16}$ im Vergleich zum Pixel-Domain Encoder.

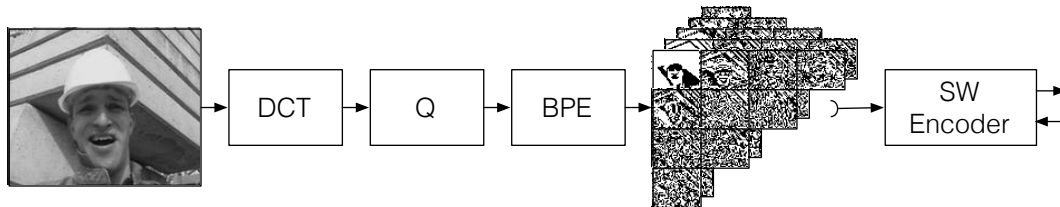
Bitebenentrennung (BPE) — Die Bitebenentrennung besitzt eine sehr geringe Komplexität, welches durch die Anzahl der Quantisierungssymbole beeinflusst wird und somit durch Verwendung der DCT leicht sinken kann.

Slepian-Wolf (SW) Encodierung — Die Komplexität der nachfolgenden binären Slepian-Wolf Encodierung hängt linear von der Anzahl zu codierender binärer Symbole ab (Abschn. 2.1.2). Für die in Abbildung 3.1 dargestellte Konfiguration bedeutet dies den gleichen Aufwand sowohl für den DCT- als auch für den Pixel-Domain Encoder. In beiden Fällen werden circa zwei binäre Symbole pro Pixel codiert.

DCT basierte DVC Codecs zeigen im Allgemeinen eine höhere RD-Performanz im Vergleich zu pixelbasierten Systemen. Um einen fairen Vergleich der Komplexität zu ermöglichen, muss dieser unter Berücksichtigung der Rekonstruktionsqua-



(a) Pixelbasierter WZ-Encoder, Anzahl zu codierender Symbole 2 bpp, $M = 2$



(b) DCT basierter WZ-Encoder, Anzahl zu codierender Symbole 1.88 bpp, Blockgröße $4 \times 4 \text{ px}^2$, $Q_i = 4$ [DISCOVER 2007]

Abbildung 3.1.: Pixel-Domain und DCT-Domain WZ-Encoder im Vergleich

lität in einem Gesamtsystem erfolgen. Festzuhalten bleibt, dass durch die zu erwartende höhere RD-Performanz eines DCT basierten Systems die Anzahl der zu codierenden Symbole sinken kann, was die Komplexität der SW-Codierung reduziert.

Die Leistungsfähigkeit der SW-Decodierung steigt mit der Länge der zu codierenden Symbolfolge an (Abb. 5.5, S. 105). Im Fall des Pixel-Domain DVC Codec entspricht die Anzahl der binären Symbole in einer Folge der Anzahl Pixel in einem Bild. Beim DCT basierten System enthalten die Bitebenen pro Frequenzband weniger Symbole, im Fall eines $4 \times 4 \text{ px}^2$ Transformation nur $\frac{1}{16}$. Diese müssen zu langen Symbolfolgen zusammengefasst werden, um keine Performanzeinbußen bei der SW-Codierung, im Vergleich zum pixelbasierten Ansatz, hervorzurufen, was zu einem erhöhten Organisationsaufwand am Encoder führt.

3.2.3. Fazit

Die bisher beschriebenen Vor- und Nachteile einer DCT basierten im Vergleich zu einer pixelbasierten Encoderlösung sind im Folgenden zusammengefasst dargestellt:

++ höhere RD-Performanz

- + leicht verringerte Komplexität der Quantisierung, da einige Bänder nicht übertragen werden
 - ähnliche Komplexität der SW-Codierung
- zusätzlicher Aufwand, um lange Symbolfolgen aus einzelnen DCT-Frequenzbändern zusammenzustellen, was notwendig zum Erreichen einer hohen Leistungsfähigkeit des SW-Codierers ist
- höhere Komplexität durch zusätzliche Transformation

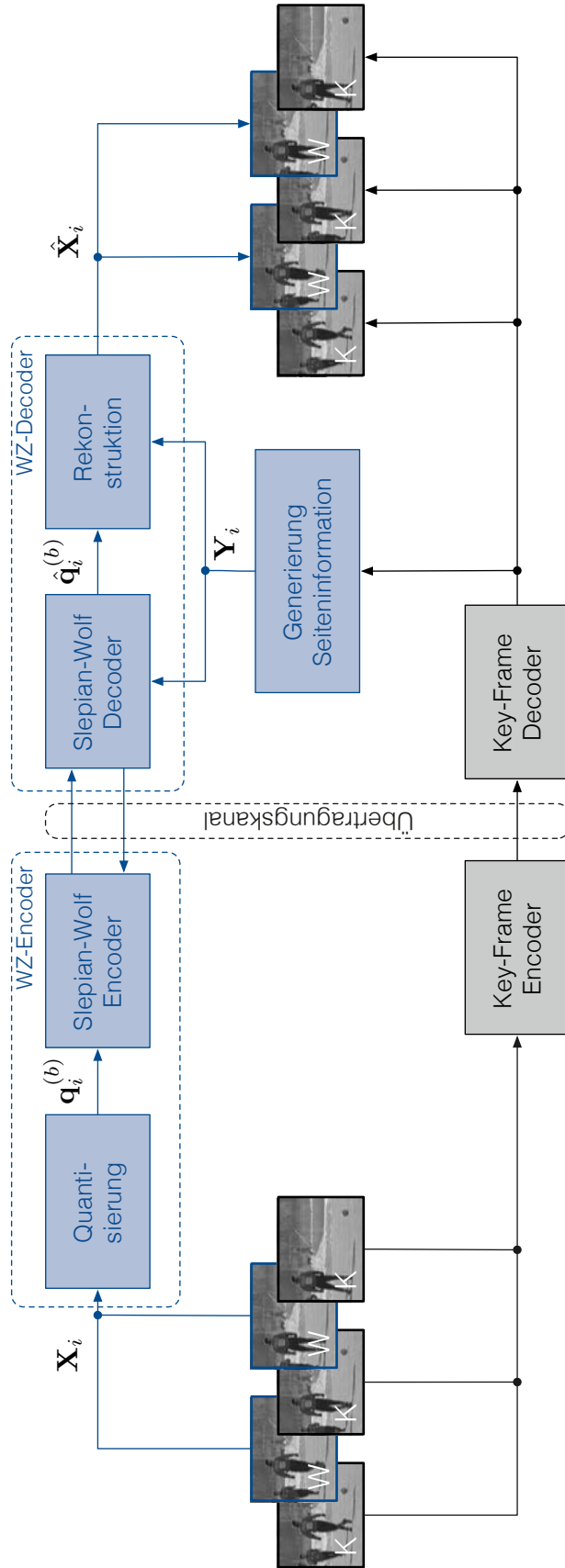
Eine wesentliche Forderung, an das im Rahmen dieser Arbeit neu entwickelte System, ist die geringe Komplexität des Encoders. Durch die Verwendung der DCT wird die Encoderkomplexität erhöht, jedoch auch die RD-Performanz verbessert. Als Konsequenz des durchgeführten Vergleichs wird in der vorliegenden Arbeit ein pixelbasiertes DVC System wegen der vergleichsweise geringen Encoderkomplexität gegenüber DCT basierten DVC Systemen favorisiert.

3.3. Grundstruktur und Komponenten

Die Diskussion im vorangegangenen Abschnitt 3.2 hat gezeigt, dass ein DVC Codec mit DCT zu einer höheren Encoderkomplexität führt. Diese Erkenntnisse führen zur Entscheidung für ein pixel basiertes Codierungssystem. Die Grundstruktur richtet sich nach der weitverbreiteten Stanford-Architektur (Abschn. 2.4.1, [Aaron u. a. 2002, 2003b]).

Zur Codierung der Videosequenzen werden zwei verschiedene Bildertypen verwendet, die Key- und Wyner-Ziv Frames. Diese werden abwechselnd codiert. Es wird nur die Helligkeitsinformation der Frames berücksichtigt, wodurch auch die Rekonstruktionsbilder keine Farbinformation enthalten. Die Beschränkung auf die Helligkeitsinformation erfolgt aus Gründen der Vereinfachung und da insbesondere durch Verwendung der Farbinformation keine Verbesserung der leistungsbestimmenden Bewegungsschätzung zu erwarten ist.

Die verwendete Grundstruktur in Abbildung 3.2 enthält folgende Komponenten:

Abbildung 3.2.: Systemstruktur des ρDeViCe (Rostocker Distributed Video Codec)

Key-Frame En-/Decoder — Die Key-Frames werden unabhängig von anderen Bildern codiert und decodiert. Dabei können konventionelle Verfahren wie JPEG, H.264intra oder HEVCintra verwendet werden. Der Einfluss des verwendeten Key-Frame Codecs wird in Abschnitt 5.3 (S. 126) diskutiert.

Wyner-Ziv En-/Decoder — Der Wyner-Ziv En- und Decoder verarbeitet die Frames nach den Prinzipien der Verteilten Videocodierung. Diese werden unabhängig von anderen Bildern codiert, jedoch unter Ausnutzung von Ähnlichkeiten zu benachbarten Bildern decodiert. In Bezug auf eine hohe RD-Performanz kommt dem WZ-Decoder eine besondere Rolle zu, welche in Kapitel 5 (S. 97) gesondert betrachtet wird.

Generierung der Seiteninformation — Die Generierung der Seiteninformation ist der zweite wichtige Einflussfaktor auf die RD-Performanz. Eine hohe Qualität der Seiteninformation führt zu einer hohen Rekonstruktionsqualität und gleichzeitig zu einer geringeren Datenrate. Die Seiteninformation wird aus den am Decoder verfügbaren decodierten Key-Frames erzeugt. Dabei stellt eine hohe Seiteninformationsqualität auch bei komplexen Bewegungsstrukturen eine große Herausforderung dar (Kap. 4, S. 41).

Um die gestellten Anforderungen bezüglich der Encoderkomplexität und Leistungsfähigkeit erfüllen zu können, erfolgt eine Konzentration auf die Entwicklung einer effizienten WZ-Codierung sowie auf Verfahren zur Generierung der Seiteninformation (Kap. 4 & 5). Die Anforderung höherer Flexibilität im Decodierungsprozess wird gesondert in Ansätzen als erweiterter Ausblick in Kapitel 7 (S. 143) untersucht.

4

Kapitel 4

Generierung der Seiteninformation – *ρDeViCe*

Dieses Kapitel beschreibt die Seiteninformationsgenerierung, welche eine der beiden wesentlichen Komponenten des entwickelten DVC Codecs *ρDeViCe* darstellt. Komplexe Bewegungsstrukturen, die für die temporale Interpolation eine große Herausforderung darstellen, bilden das Kernproblem für die Generierung der Seiteninformation.

Die Generierung der Seiteninformation ist wegen ihres starken Einflusses auf die Leistungsfähigkeit des Gesamtsystems von großer Bedeutung. Wie in den Grundlagen (Abschn. 2.4.2) beschrieben, können unterschiedliche Abhängigkeiten in einer Videosequenz ausgewertet werden, um die Seiteninformation zu generieren. In dieser Arbeit erfolgt eine Beschränkung auf die Ermittlung der Seiteninformation auf Basis von zeitlichen Korrelationen, also durch zeitliche Inter-/Extrapolation. In der Literaturrecherche (Abschn. 2.4.2) hat sich gezeigt, dass Verfahren, welche die zeitliche Korrelation nutzen, eine höhere Qualität der Seiteninformation im Vergleich zu anderen verfügbaren Verfahren liefern.

In diesem Abschnitt werden zunächst ausgewählte zeitliche Interpolationsverfahren aus dem Bereich der DVC im Detail beschrieben, um dann darauf aufbauend die im Rahmen dieser Arbeit entwickelten Verfahren vorzustellen. Diese Verfahren konzentrieren sich auf die Bewegungsformen schnelle Kamerabewegung und bewegte Objekte (Anhang A.2.1, S. 199), welche durch einfache zeitliche Interpolation nicht abgedeckt werden können.

4.1. Ausgewählte bekannte Verfahren zur Generierung der Seiteninformation

Die bislang bekannten Verfahren zur Generierung der Seiteninformation unter Ausnutzung der zeitlichen Korrelation wurden im Abschnitt 2.4.2 in mehrere Gruppen eingeteilt. Hier werden kurz einige Vertreter der Blockmatching Verfahren im Detail vorgestellt, welche als Grundlage für weitere neu entwickelte Verfahren in dieser Arbeit dienen (Blockmatching siehe auch Anhang A.2.1, S. 198).

Beim Blockmatching wird das Bild in aneinandergrenzende Blöcke zerlegt und jedem Block ein Bewegungsvektor zugeordnet. Dieser wird durch Vergleich des aktuellen Blockes mit mehreren Blöcken im Referenzbild bestimmt. Die Blöcke müssen dabei innerhalb des Suchfensters liegen. Ein großes Suchfenster erhöht die Komplexität dabei stark. Als Ähnlichkeitsmaß wird die mittlere absolute Distanz (MAD – *mean absolute difference*) oder der mittlere quadratische Fehler (MSE – *mean square error*) verwendet.

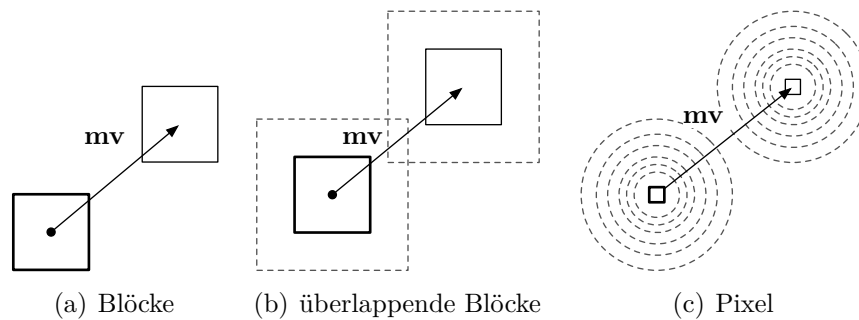


Abbildung 4.1.: Unidirektionale Bewegungsschätzung (a) für aneinandergrenzende Blöcke, (b) für überlappende Blöcke und (c) auf Pixelbasis

Neben dem Blockmatching auf Basis aneinandergrenzender Blöcke (Abb. 4.1(a)), wird in der Literatur auch das Blockmatching mit überlappenden benachbarten Blöcken (Abb. 4.1(b)) vorgeschlagen. Der Vorteil hierbei ist, dass durch die größere Fläche der Blöcke eine sicherere Ermittlung der wahren Bewegung erfolgen kann. Des Weiteren bleiben das Blockraster und somit auch das Bewegungsvektorfeld ähnlich dicht wie bei dem einfachen Blockmatching.

Ein sehr dichtes Bewegungsvektorfeld wird durch ein pixelbasiertes Blockmatching erreicht, indem für jedes Pixel (Blockgröße $1 \times 1 \text{ px}^2$) ein modifiziertes Blockmat-

ching, basierend auf dessen gewichteter Nachbarschaft (wSAD - *weighed sum of absolute difference*), durchgeführt wird (Abb. 4.1(c)). Dies kann auch als Blockmatching mit überlappenden Blöcken interpretiert werden, wobei der Überlappungsgrad sehr groß und das Blockraster sehr klein ist.

Wichtige Verfahren, basierend auf einfachem Blockmatching, sind BiMESS (*Bidirectional Motion Estimation and Spatial Smoothing*, [Ascenso u. a. 2005]) und dessen Weiterentwicklung MCTI (*Motion Compensated Temporal Interpolation*, [Ascenso u. Pereira 2008]). Ein Vertreter des pixelbasierten Blockmatchings stellt das PBTI (*Pixel-based Temporal Interpolation*, [Sofke u. a. 2009]) dar.

Die genannten Verfahren BiMESS und PBTI liefern die Basis für die im Rahmen dieser Arbeit entwickelten Verfahren. Einen Überblick der in dieser Arbeit neu entwickelten Verfahren und deren Abhängigkeit von den bekannten Verfahren zeigt Abbildung 4.5 (S. 47).

4.1.1. BiMESS – *Bidirectional Motion Estimation and Spatial Smoothing* & MCTI – *Motion Compensated Temporal Interpolation*

Bei den Verfahren BiMESS [Ascenso u. a. 2005] und MCTI [Ascenso u. Pereira 2008] handelt es sich um Methoden zur Generierung der Seiteninformation durch zeitliche Interpolation, wobei eine blockbasierte Bewegungskompensation durchgeführt wird. In Abbildung 4.2 ist das kombinierte Blockschaltbild für beide Methoden dargestellt. Zwei benachbarte Key-Frames $\mathbf{K}_{i-1}, \mathbf{K}_{i+1}$ werden verwendet, um für den zwischenliegenden Wyner-Ziv Frame eine initiale Schätzung der Seiteninformation \mathbf{Y}_i zu erzeugen.

Im Falle von BiMESS wird hierzu zunächst eine Vorwärtsbewegungsschätzung (FME – *forward motion estimation*, Blockgröße $8 \times 8 \text{ px}^2$) auf Basis der tiefpassgefilterten Key-Frames durchgeführt [Ascenso u. a. 2005]. Das so erhaltene Bewegungsvektorfeld wird als Startpunkt für die nachfolgende bidirektionale Bewegungsschätzung (BiME – *bidirectional motion estimation*) verwendet, wodurch ein auf die Seiteninformation zentriertes Bewegungsvektorfeld erzeugt wird. Die örtliche Abhängigkeit der Bewegungsvektoren wird durch die Vektormedianfilterung ausgenutzt, wodurch das Bewegungsvektorfeld geglättet wird (SMS – *spatial motion smoothing*). Den

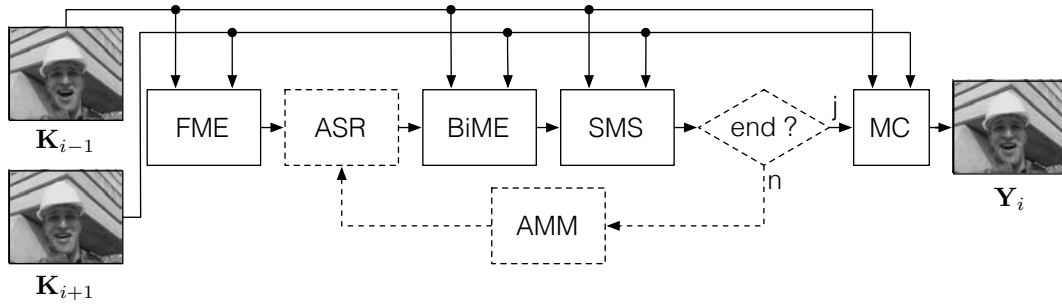


Abbildung 4.2.: Seiteninformation durch temporale Interpolation mit BiMESS [Ascenso u. a. 2005] bzw. MCTI [Ascenso u. Pereira 2008] (gestrichelte Linien nur in MCTI)

Abschluss bildet die Bewegungskompensation (MC – *motion compensation*) bei der benachbarter Key-Frames, wobei die Seiteninformation aus dem Mittelwert der beiden bewegungskompensierten Bilder gebildet wird.

Die MCTI Methode [Ascenso u. Pereira 2008] stellt eine Erweiterung des BiMESS Verfahrens dar. Im Wesentlichen handelt es sich um ein hierarchisches blockbasiertes Verfahren, bei dem Blockgrößen von $16 \times 16 \text{ px}^2$, $8 \times 8 \text{ px}^2$ und $4 \times 4 \text{ px}^2$ eingesetzt werden. Den Startpunkt bildet auch hier eine Vorwärtsbewegungsschätzung (FME) mit großer Blockgröße und großem Suchfenster. Anschließend wird das Bewegungsvektorfeld iterativ durch bidirektionale Bewegungsschätzung und Vektormedianfilterung verfeinert. Dabei werden immer kleinere Blockgrößen, eine adaptive Anpassung des Suchfensters (ASR – *adaptive search range*) und ein affines Bewegungsmodell (AMM – *affine motion model*) eingesetzt. Die Seiteninformation wird, in Anlehnung an BiMESS, durch Mittelung der bewegungskompensierten Key-Frames erzeugt.

4.1.2. PBTI – *Pixel-based Temporal Interpolation*

Im Gegensatz zu den bisher beschriebenen blockbasierten Verfahren setzt PBTI (*Pixel-based Temporal Interpolation*, [Sofke u. a. 2009][Sofke 2009]) eine pixelbasierte Bewegungsschätzung und -kompensation ein (Abb. 4.3). Im ersten Schritt wird sowohl eine pixelbasierte Vorwärts- als auch eine Rückwärtsbewegungsschätzung (pFME, pBME) zwischen den beiden Key-Frames durchgeführt. Hierbei wird der

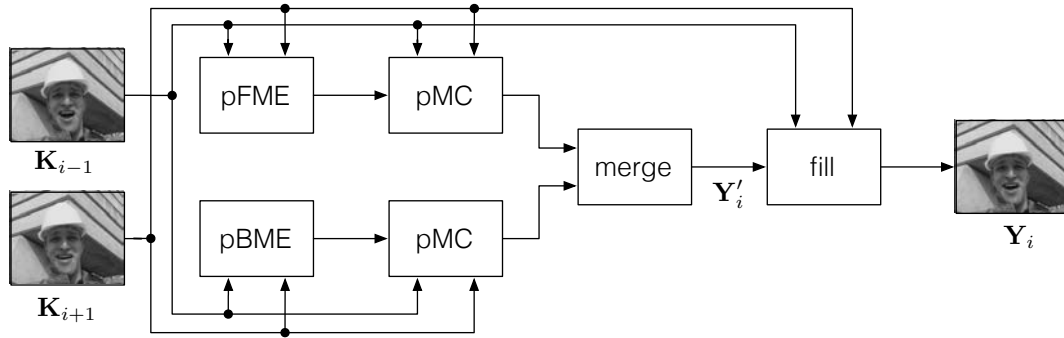


Abbildung 4.3.: Seiteninformationsgenerierung durch temporale Interpolation mit PBTi [Sofke u. a. 2009][Sofke 2009]

SAD, der mit einem Gauß-Fenster gewichteten Nachbarschaft, als Kostenfunktion für das Blockmatching verwendet. Eine Bewegungskompensation wird für beide Key-Frames jeweils mit beiden Bewegungsvektorfeldern durchgeführt. Anschließend wird über alle vier bewegungskompensierten Frames gemittelt (merge), um die vorläufige Seiteninformation Y'_i zu erhalten.

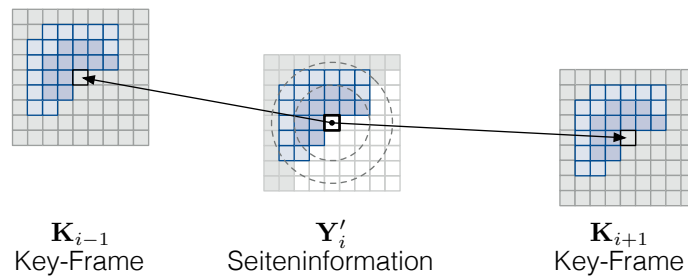


Abbildung 4.4.: Füllen (fill) von Lücken in der Seiteninformation Y'_i durch Bewegungsschätzung mit gewichteten Nachbarpixeln bei PBTi (graue und farbige Pixel sind bekannt)

Da aufgrund inhomogener Bewegung sowie Auf- und Verdeckungen einige Bildbereiche nicht gefüllt sind, werden diese Lücken abschließend durch eine Bewegungsschätzung und -kompensation unter Zuhilfenahme der bereits bekannten Nachbarpixel geschätzt (fill, blaue Pixel, siehe Abb. 4.4). Die Nachbarschaft wird dabei mit einem Gauß-Fenster gewichtet.

4.1.3. Bewertung der bekannten Verfahren

Die blockbasierten Verfahren BiMESS und MCTI neigen zur Bildung von Blockartefakten. Dies ist aufgrund der Blockgröße von $8 \times 8 \text{ px}^2$ bei BiMESS stärker ausgeprägt als bei MCTI, welches einen hierarchischen Ansatz verwendet (minimale Blockgröße $4 \times 4 \text{ px}^2$). Das Problem der Blockartefakte tritt bei PBTI nicht auf, da hier ein dichtes Bewegungsvektorfeld bestimmt wird.

Starke Bewegungen können zur fehlerhaften Bestimmung von Bewegungsvektoren mittels Blockmatching führen. Dieses Problem wird in MCTI durch ein hierarchisches Blockmatching (maximale Blockgröße $16 \times 16 \text{ px}^2$) und durch eine örtliche Glättung des Bewegungsvektorfeldes, welche auch in BiMESS verwendet wird, gelöst.

Eine gute zeitliche Interpolation kann erreicht werden, wenn das verwendete Verfahren

1. über eine robuste Bewegungsschätzung verfügt,
2. den örtlichen Zusammenhang der Bewegungsvektorfelder ausnutzt und
3. Blockartefakte vermeidet.

Diese Erkenntnisse sollen in die Entwicklung neuer Verfahren einfließen. Weiterhin offene Probleme, wie

- die Bewegungsschätzung bei sehr starker Bewegung und
- die zeitliche Interpolation in Auf- oder Verdeckungsbereichen,

sollen durch die neu entwickelten Verfahren gelöst werden.

4.2. Übersicht der entwickelten Verfahren zur Generierung der Seiteninformation

Abbildung 4.5 gibt einen zusammenhängenden Überblick über die aus der Literatur bekannten Verfahren und den darauf aufbauenden neu entwickelten Verfahren. Diese werden beginnend mit basicPBTI, welches eine vereinfachte Version des PBTI darstellt, nachfolgend beschrieben. Es werden aufeinander aufbauend die Probleme starke globale Bewegung (GTIE, GpTIE, Abschn. 4.4), inhomogene Bewegung

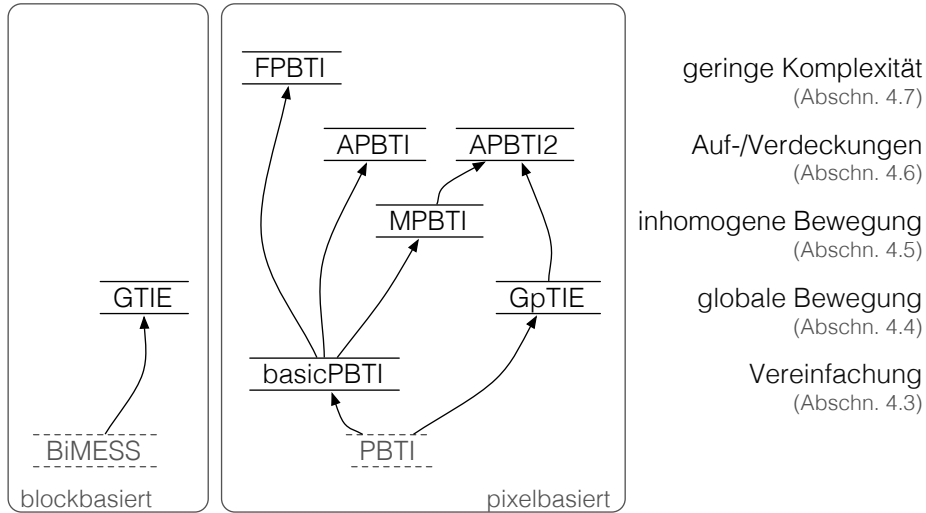


Abbildung 4.5.: Übersicht der entwickelten Verfahren zur Generierung der Seiteninformation (schwarz) und aus der Literatur bekannter Basisverfahren (grau)

(MPBTI, Abschn. 4.5) und das Problem der Auf- und Verdeckungen (APBTI, APBTI2, Abschn. 4.6) bearbeitet. Zusätzlich werden abschließend Ansätze zur Verringerung der Komplexität vorgestellt (FPBTI, Abschn. 4.7).

4.3. Vereinfachte pixelbasierte temporale Interpolation (*basicPBTI*)

Den Ausgangspunkt für viele im Rahmen dieser Arbeit entwickelte Verfahren (MPBTI, APBTI, APBTI2, FPBTI) stellt die im Folgenden beschriebene vereinfachte Version *basicPBTI* des PBTI Verfahrens [Sofke u. a. 2009] dar (Abb.4.5). Die pixelbasierte Vorwärts- und Rückwärtsbewegungsschätzung wurde durch eine Vorwärtsbewegungsschätzung (pFME) ersetzt (Abb. 4.6). Weiterhin wird eine bidirektionale pixelbasierte Bewegungsschätzung (pBiME) zur Verfeinerung des Bewegungsvektorfeldes eingesetzt, wodurch keine Lücken durch inhomogene Bewegung entstehen und so kein Füllalgorithmus notwendig ist. Den Abschluss bildet auch hier die Bewegungskompensation und anschließende Mittelung beider Key-Frames.

Die *basicPBTI* Methode wird hier im Detail beschrieben, um ein tief gehendes

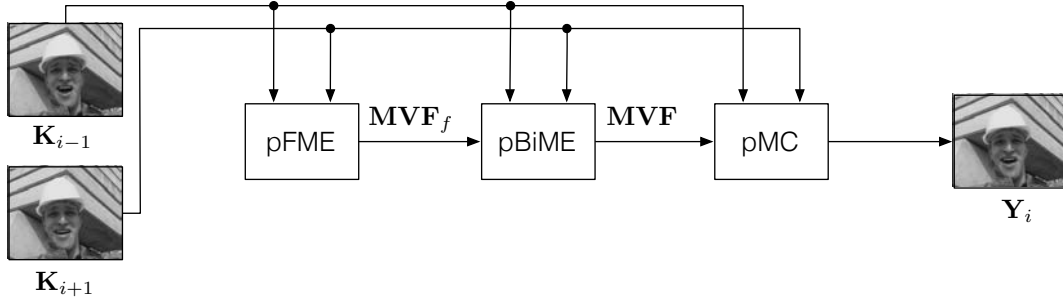


Abbildung 4.6.: basicPBTi als vereinfachte Version des PBTi [Sofke u. a. 2009]

Verständnis für die im weiteren Verlauf vorgeschlagenen Verfahren zu ermöglichen. Den ersten Schritt stellt die pixelbasierte Vorwärtsbewegungsschätzung (pFME) von Key-Frame \mathbf{K}_{i-1} nach \mathbf{K}_{i+1} dar (Gl. 4.1). Ausgehend von einem Pixel im vorangegangenen Key-Frame \mathbf{K}_{i-1} an der Position (x_0, y_0) wird dessen Korrespondenz $(x_0 + mv_{f,x}, y_0 + mv_{f,y})$ im nachfolgenden Key-Frame \mathbf{K}_{i+1} bestimmt. Dazu wird die Nachbarschaft zum modifizierten Blockmatching mittels gewichteten SAD verwendet. Zur Wichtung wird ein Gauß-Fenster mit dem Radius $\sigma = 5$ px eingesetzt (Gl. 4.2). Weiterhin wird der SAD nur innerhalb des Prüfensters MW (*matching window*) ausgewertet (Gl. 4.6), da die Gewichte $\mathbf{GW}_\sigma(k, l)$ außerhalb dieses Fensters klein sind. Der Suchbereich SW_f (*search window*) für die Vorwärtssuche ist dabei mit $SR_f = 10$ px so gewählt, dass ein guter Kompromiss zwischen dem Erfassen starker Bewegungen und möglichen Fehlschätzungen der Bewegung gewährleistet wird. Diese Bewegungsschätzung wird aus Komplexitätsgründen mit einer Genauigkeit von ganzen Pixeln (\mathbb{Q}_1) durchgeführt.

$$\mathbf{mv}_f = \arg \min_{mv_{f,x}, mv_{f,y}} \sum_{(k,l) \in MW} \mathbf{GW}_\sigma(k, l) \cdot \left| \mathbf{K}_{i-1}(x_0 + k, y_0 + l) - \mathbf{K}_{i+1}(x_0 + k + mv_{f,x}, y_0 + l + mv_{f,y}) \right| \quad (4.1)$$

$$\mathbf{GW}_\sigma(k, l) = \frac{1}{2\pi\sigma^2} e^{-\frac{k^2 + l^2}{2\sigma^2}} \quad (4.2)$$

$$\mathbf{mv}_f \in SW_f \quad (4.3)$$

$$SW_f = \left\{ (mv_x, mv_y) \in \mathbb{Q}_1^2 \mid -SR_f \leq mv_x \leq SR_f \wedge -SR_f \leq mv_y \leq SR_f \right\} \quad (4.4)$$

$$\mathbb{Q}_b = \left\{ q \in \mathbb{Q} \mid \dots - \frac{2}{b}; -\frac{1}{b}; 0; \frac{1}{b}; \frac{2}{b}; \dots \right\} \quad (4.5)$$

$$MW = \{(k, l) \in \mathbb{Q}_1^2 \mid -10 \leq k \leq 10 \wedge -10 \leq l \leq 10\} \quad (4.6)$$

Der so geschätzte Bewegungsvektor $\mathbf{mv}_f = (mv_{f,x}, mv_{f,y})$ ist Teil des Bewegungsfeldes $\mathbf{MVF}_f(x_0, y_0) = \mathbf{mv}_f$ und wird für jedes Pixel des Key-Frames \mathbf{K}_{i-1} bestimmt.

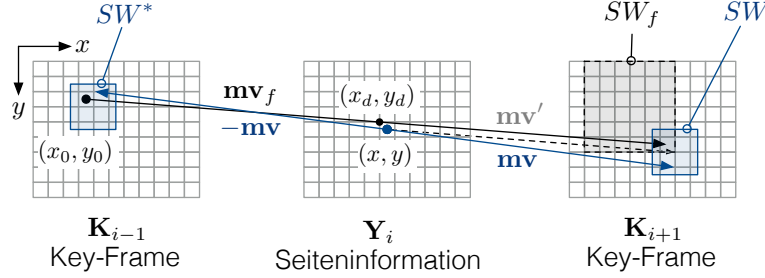


Abbildung 4.7.: Pixelbasierte Vorwärtsbewegungsschätzung schwarz und nachfolgende bidirektionale Bewegungsschätzung blau

Im folgenden Schritt wird jedem Pixel (x, y) der Seiteninformation \mathbf{Y}_i ein Bewegungsvektor $\mathbf{mv}' = \mathbf{MVF}'(x, y)$ zugeordnet. Dies geschieht, indem der Bewegungsvektor \mathbf{mv}_f ausgewählt wird, dessen Durchstoßpunkt (x_d, y_d) am dichtesten an der Position des aktuellen Pixel (x, y) liegt (Gl. 4.7). Aufgrund der verkürzten zeitlichen Distanz zwischen Seiteninformation und Key-Frame wird die Länge des Bewegungsvektors \mathbf{mv}' entsprechend auf die Hälfte verkürzt.

$$\mathbf{mv}' = \frac{1}{2} \arg \min_{mv_{f,x}, mv_{f,y}} \left\{ \left(\underbrace{x_0 + \frac{mv_{f,x}}{2}}_{x_b} - x \right)^2 + \left(\underbrace{y_0 + \frac{mv_{f,y}}{2}}_{y_b} - y \right)^2 \right\} \quad (4.7)$$

Der so ermittelte Bewegungsvektor \mathbf{mv}' dient als Startpunkt für die pixelbasierte bidirektionale Bewegungsschätzung (Gl. 4.8), welche zur Verfeinerung des Bewegungsfeldes mit einer Halb-Pixelgenauigkeit durchgeführt wird. Der Suchradius ist mit $SR = 1 \text{ px}$ wesentlich eingeschränkt, wobei das Suchfenster um den Bewegungsvektor \mathbf{mv}' zentriert wird.

Die für die Halb-Pixel genaue Bewegungsschätzung notwendigen Zwischenpositionen werden durch ein lineares zweidimensionales Interpolationsfilter erzeugt. Wei-

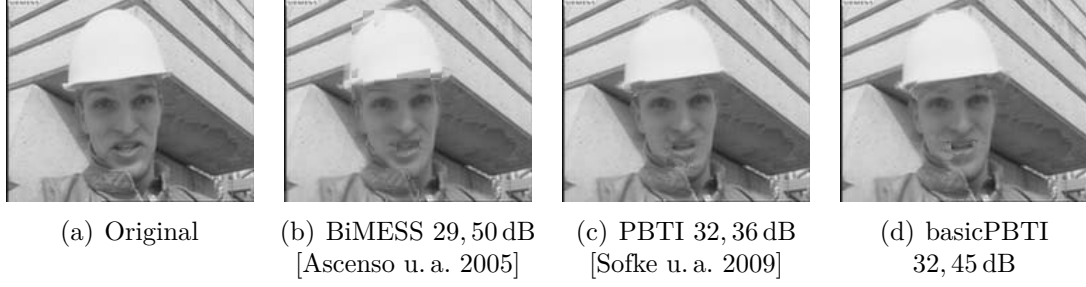


Abbildung 4.8.: Vergleich der Seiteninformationsqualität (PSNR-Y), Foreman, QCIF, 30 fps, WZ-Frame 72

terhin wird eine konstante Erweiterung der Randbereiche verwendet.

$$\mathbf{mv}(x_0, y_0) = \arg \min_{mv_x, mv_y} \sum_{(k,l) \in MW} \mathbf{GW}_\sigma(k, l) \cdot |\mathbf{K}_{i-1}(x_0 + k - mv_x, y_0 + l - mv_y) - \mathbf{K}_{i+1}(x_0 + k + mv_x, y_0 + l + mv_y)| \quad (4.8)$$

$$\mathbf{mv} \in SW \quad (4.9)$$

$$SW = \left\{ (mv_x, mv_y) \in \mathbb{Q}_2^2 \mid -SR + mv'_x \leq mv_x \leq SR_f + mv'_x \wedge -SR + mv'_y \leq mv_y \leq SR + mv'_y \right\} \quad (4.10)$$

Abschließend wird die Bewegungskompensation gemäß Gleichung 4.11 durchgeführt und die Seiteninformation durch Mittelung der bewegungskompensierten Key-Frames gewonnen.

$$\mathbf{Y}_i(x, y) = \frac{1}{2} \left\{ \mathbf{K}_{i-1}^{-\mathbf{MVF}}(x, y) + \mathbf{K}_{i+1}^{\mathbf{MVF}}(x, y) \right\} \quad (4.11)$$

In Abbildung 4.8 wird das basicPBTi Verfahren mit den zuvor vorgestellten Methoden BiMESS und PBTi verglichen. Festzuhalten ist, dass, wie bei PBTi auch, der Einfluss von Blockartefakten stark reduziert wird. Weiterhin liegt die objektive Qualität auf ähnlichem Niveau wie PBTi, wobei die subjektive Qualität etwas abnimmt.

4.4. Globale Bewegung

Im Abschnitt 4.1 wurden beispielhaft block- und pixelbasierte Verfahren zur Bewegungsschätzung und -kompensation vorgestellt. Für diese auf Blockmatching basierenden Verfahren stellen sehr starke Bewegungen (lange Bewegungsvektoren) ein großes Problem dar. Im Folgenden Abschnitt werden die Problemstellung schneller Bewegung beschrieben und darauf aufbauend neue Verfahren zur Lösung dieser Herausforderung beschrieben.

Die Erfassung schneller Bewegungen erfordert die starke Vergrößerung des Suchfensters, was im Wesentlichen zu zwei Problemen führt. Zum einen steigt die Wahrscheinlichkeit, dass durch das Blockmatching ein Bewegungsvektor bestimmt wird, welcher nicht der wahren Bewegung entspricht. Dies wird durch die größere Vielfalt an Blöcken und Bewegungsvektoren hervorgerufen. Zum anderen steigt der notwendige Rechenaufwand proportional mit der Größe des Suchfensters an.

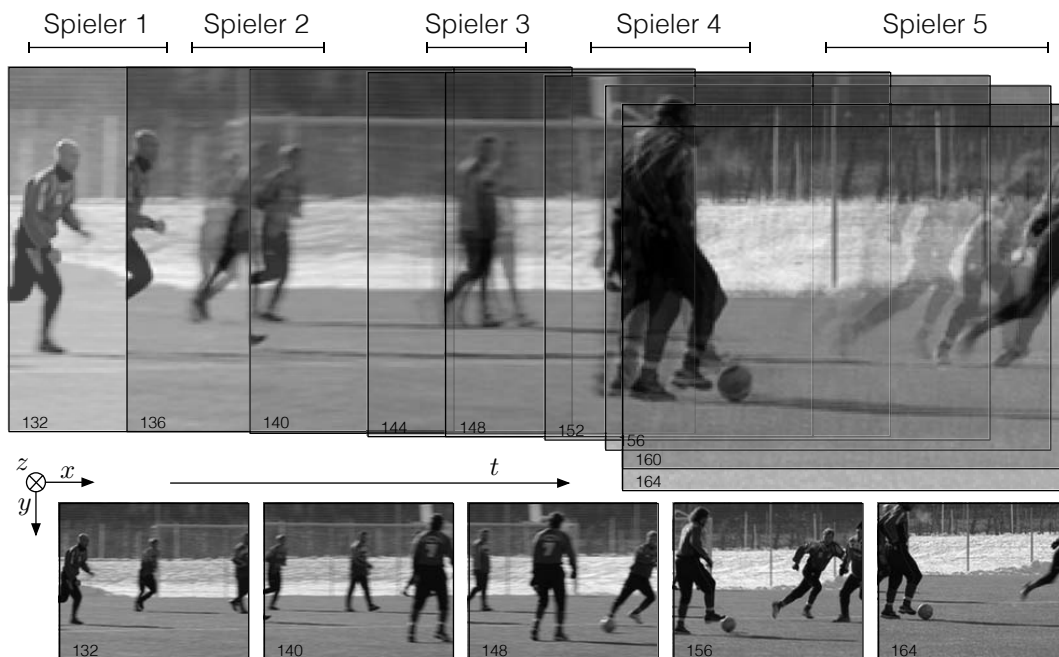


Abbildung 4.9.: Dominante globale Bewegung in der Sequenz **Soccer**, Frame 132 bis 164, (oben) jedes 4. Bild anhand des Hintergrundes aufeinander ausgerichtet, (unten) exemplarische Bilder der Szene

Starke Bewegung kann zum einen die Folge von sich schnell bewegendenden Objekten sein. Zum anderen kann dies auch durch eine schnelle Bewegung der Kamera, wie

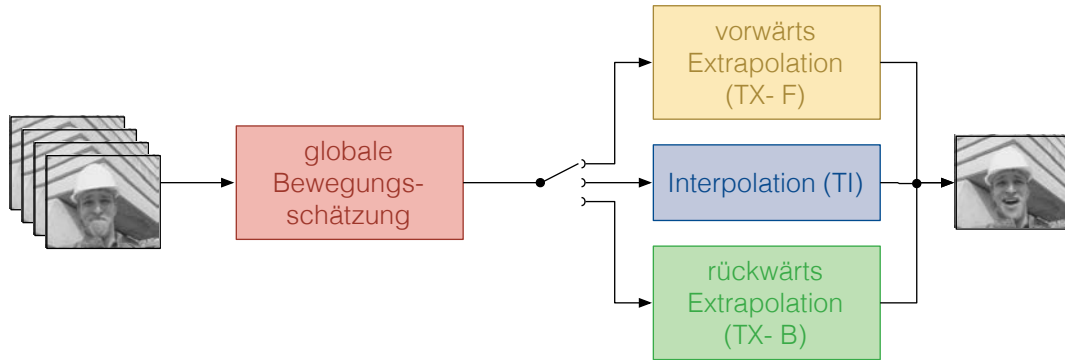


Abbildung 4.10.: Vereinfachtes Blockschaltbild der Verfahren GTIE/GpTIE

zum Beispiel durch Kameraschwenks, hervorgerufen werden. Abbildung 4.9 zeigt beispielhaft die Frames 132 bis 164 der Sequenz **Soccer**. Die Bilder sind anhand des Hintergrundes aufeinander ausgerichtet. Gut zu erkennen ist, dass die Spieler 3 und 4 sich nur wenig bewegen, diese aber durch den schnellen Kameraschwenk als stark bewegte Objekte in der Videosequenz wahrgenommen werden. Somit ruft die schnelle globale Bewegung (Kameraschwenk) eine sehr starke Bewegung von relativ zum Hintergrund unbewegten Objekten hervor.

In diesem Abschnitt werden zwei Verfahren (GTIE/GpTIE – *Global Motion Guided (Pixel-based) Adaptive Temporal Inter-/Extrapolation*) zur Lösung des Problems der starken globalen Bewegung vorgeschlagen, welche sich der zeitlichen Interpolation beim Vorhandensein von starker globaler Bewegung widmen.

Die Kernidee besteht darin, dass die globale Bewegung in einem ersten Schritt robust bestimmt und im Folgenden die zeitliche Interpolation durch die bekannte globale Bewegung geführt wird (Abb. 4.10). Dies hat den Vorteil, dass der Suchbereich für das modifizierte Blockmatching wesentlich verkleinert werden kann, wodurch der Rechenzeitbedarf und auch die Wahrscheinlichkeit einer Fehlschätzung der Bewegung wesentlich sinken. Die robuste globale Bewegungsschätzung erfolgt dabei mittels charakteristischer Punkte im Bild (Anhang A.2.2, S. 205), welche sich sehr gut wiederfinden lassen und somit eine Bestimmung der wahren Bewegung ermöglichen.

Da durch die starke Bewegung der Kamera ständig Bereiche am Bildrand neu aufgedeckt oder verdeckt werden, tritt ein großflächiges Auf-/Verdeckungsproblem auf. Dieses Problem wird durch den Einsatz temporaler Extrapolation (vorwärts und

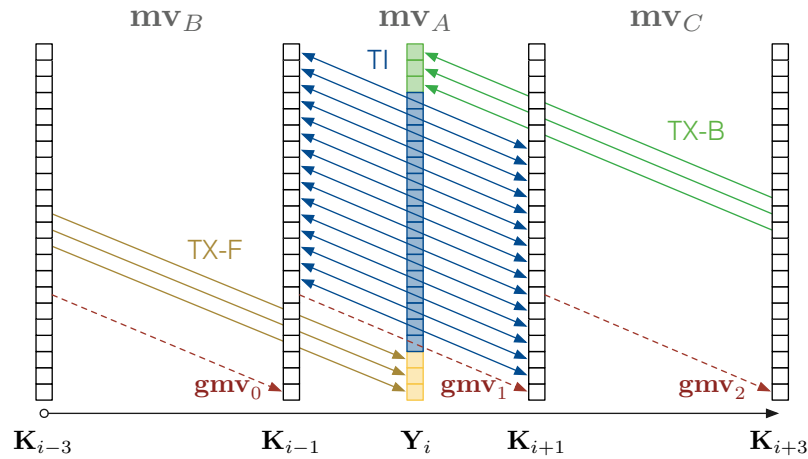


Abbildung 4.11.: Bewegungsvektorfeld unter Zuhilfenahme von globalen Bewegungsvektoren \mathbf{gmV}_j in GTIE und GpTIE, temporale Interpolation (TI) im Bild und temporale Extrapolation (TX-F/B) in Randbereichen

rückwärts) in den Randbereichen gelöst (Abb. 4.11). Dies ist zielführend, da die leistungsfähige Interpolation nur dann gute Ergebnisse liefert, wenn die Bildinhalte in beiden benachbarten Frames sichtbar sind, was in den Randbereichen bei starker Kamerabewegung nicht der Fall ist.

4.4.1. Schätzung der globalen Bewegung

Die Schätzung der globalen Bewegung stellt sowohl die Grundlage für die nachfolgende block- bzw. pixelbasierte Bewegungsschätzung als auch für die adaptive Umschaltung zwischen den Inter-/Extrapolationsmodi dar. Dabei werden folgende Anforderungen an die Schätzung der globalen Bewegung gestellt:

hohe Zuverlässigkeit — Robustheit und Zuverlässigkeit spielt eine große Rolle, da dies durch nachfolgende Verarbeitungsstufen nicht ausgeglichen werden kann

hinreichende Genauigkeit — eine hohe Genauigkeit spielt eine untergeordnete Rolle, da die nachfolgende Bewegungsschätzung Ungenauigkeiten ausgleichen kann

Die hohe Robustheit der globalen Bewegungsschätzung steht hier im Fokus. Aus diesem Grund werden folgende Überlegungen angestellt, um die Robustheit sicherzustellen:

einfaches Modell — Beschreibung der globalen Bewegung durch ein einfaches 2-parametrisches Modell, um den Schätzfehler der Modellparameter zu verringern

markante Bildbereiche — globale Bewegung bestimmen, indem die Bewegung ausschließlich nur von charakteristischen Bereichen berücksichtigt wird

Wichtung — die Bewegung der charakteristischen Bildbereiche wird mit der Größe ihres Umgebungsbereiches gewichtet, um großflächige homogene Bereiche nicht gegenüber stark strukturierten Bereichen zu benachteiligen

Entfernung von Ausreißern — Ausreißer werden mittels gewichteter Vektormedianfilterung entfernt

Die globale Bewegung kann mit komplexen parametrischen Modellen abgebildet werden, welche auch Scherung und Rotation berücksichtigen (Abschn. A.2.1, S. 200). Für den hier betrachteten Anwendungsfall steht die Robustheit und hinreichende Genauigkeit im Vordergrund. Aus diesem Grund wird die globale Bewegung zwischen einem Punkt (x, y) im Frame \mathbf{F} und einem Punkt (x', y') im Frame \mathbf{F}' durch ein einfaches 2-parametrisches Modell beschrieben (Gl. 4.12). Dieses Modell erfasst nur eine Translation und kann somit keine Rotation um die z -Achse oder perspektivische Verzerrungen abbilden (Koordinatenachsen, siehe Abb. 4.9).

Eine schnelle Kamerabewegung wird meist durch einen Schwenk (Rotation um die y -Achse) hervorgerufen. Dieser führt zu einer starken horizontalen Verschiebung und einer perspektivischen Verzerrung. Letztere hat geringen Einfluss auf die Bewegung im Bild, womit Rotationen um die y - oder auch x -Achse mit hinreichender Genauigkeit durch dieses Modell abgebildet werden können.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & gmv_x \\ 0 & 1 & gmv_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (4.12)$$

Die Modellparameter werden als globaler Bewegungsvektor $\mathbf{gmv} = (gmv_x, gmv_y)$ zusammengefasst und aus der Bewegung markanter Bereiche im Bild bestimmt. Diese Bewegung wird in einem mehrstufigen Prozess (Abb. 4.12) ermittelt:

1. Finden markanter Punkte (*Feature Point Detection*)
2. Berechnung eines Merkmalsvektors (*Feature Extraction*)
3. Zuordnung von Merkmalsvektoren (*Matching*)
4. Bestimmung der Bewegung der Merkmalspunkte
5. Wichtung der Merkmalspunkte anhand der Umgebungsfläche
6. Medianfilterung

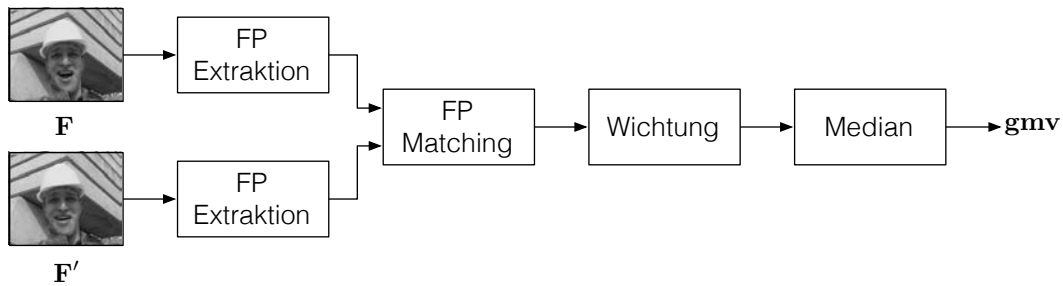


Abbildung 4.12.: Bestimmung des globalen Bewegungsvektors auf Basis der Zuordnung von charakteristischen Punkten (FP - *feature point*)

Für die aufgeführten Schritte (1-3) wird das SIFT-Verfahren (*Scale Invariant Feature Transform*, [Lowe 2004], [Vedaldi u. Fulkerson 2012]) verwendet. Eine detaillierte Beschreibung der SIFT- und weiterer Verfahren ist im Anhang A.2.2 zu finden.

Die Zuordnung von Merkmalsvektoren findet dabei gerichtet statt. Es wird versucht, jedem Merkmalsvektor im Frame \mathbf{F} einen Merkmalsvektor im Frame \mathbf{F}' zuzuordnen (Abb. 4.13). Somit wird eine Menge von Paarungen $(fp, fp') \in M_1$ gefunden. Es ist dabei nicht zwingend erforderlich, dass jeder Merkmalsvektor Teil einer Paarung ist. Weiterhin können einem Repräsentanten fp' auch mehrere Feature-Punkte fp zugeordnet werden. Die so erhaltene Zuordnung ist somit nicht eineindeutig (injektiv). Solch eine Zuordnung ist ein Anzeichen für eine schwache und somit wenig robuste Paarung. Dieses Problem wird gelöst, indem zusätzlich eine rückwärtsgerichtete Zuordnung $(fp', fp) \in M_2$ durchgeführt wird (jedem fp' wird ein fp zugeordnet). Weiterführend werden nur diejenigen Paarungen M zur Bewegungsbestimmung verwendet, welche sowohl Teil von M_1 als auch M_2 sind (Gl. 4.13,

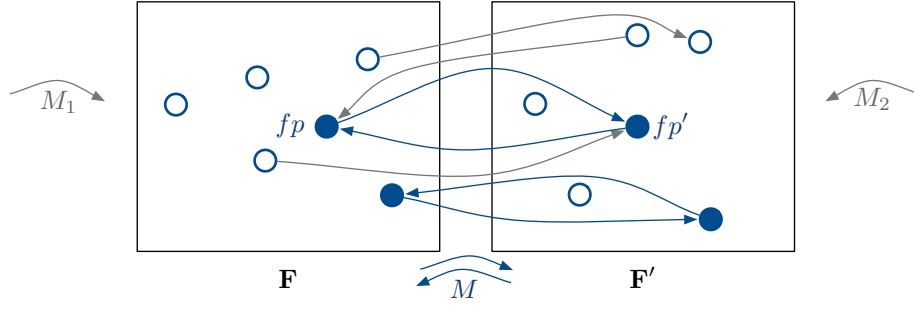


Abbildung 4.13.: Paarung von Merkmalsvektoren zwischen den Frames \mathbf{F} und \mathbf{F}' , ausschließlich injektive (eindeutige) Paarungen (blau) werden verwendet

Abb. 4.13).

$$M = M_1 \cap M_2 \quad (4.13)$$

Jeder Paarung $(fp, fp') \in M$ wird ein Bewegungsvektor \mathbf{mv} zugeordnet (Gl. 4.14). Aus dieser Menge von Bewegungsvektoren wird nachfolgend durch gewichtete Medianfilterung ein Vektor, der die globale Bewegung repräsentiert, ausgewählt.

$$\begin{aligned} \mathbf{mv} &= (mv_x, mv_y) = (fp'_x - fp_x, fp'_y - fp_y) \\ &\quad (fp, fp') \in M \end{aligned} \quad (4.14)$$

Der globale Bewegungsvektor \mathbf{gmv} soll die Bewegung der Mehrzahl der Pixel im Bild robust wiedergeben. Aus diesem Grund wird ein Vektormedianfilter eingesetzt, um Außreißer zu entfernen. Weiterhin wird eine Wichtung der Bewegungsvektoren durchgeführt. Hierbei wird die umgebende Fläche eines Merkmalspunktes fp als Gewicht verwendet. In homogenen Bereichen eines Bildes finden sich weniger markante Punkte, die eine zuverlässige Paarung ermöglichen. Um somit stark strukturierte Bereiche nicht durch eine höhere Anzahl an Merkmalspunkten zu bevorteilen, wird die Größe der umgebenden Fläche als Gewicht eingesetzt (Abb. 4.14).

Die umgebende Fläche eines Merkmalspunktes fp_j wird bestimmt, indem mittels Delaunay Triangulation [de Berg u. a. 2008, S.193] ein Dreiecksnetz T über alle

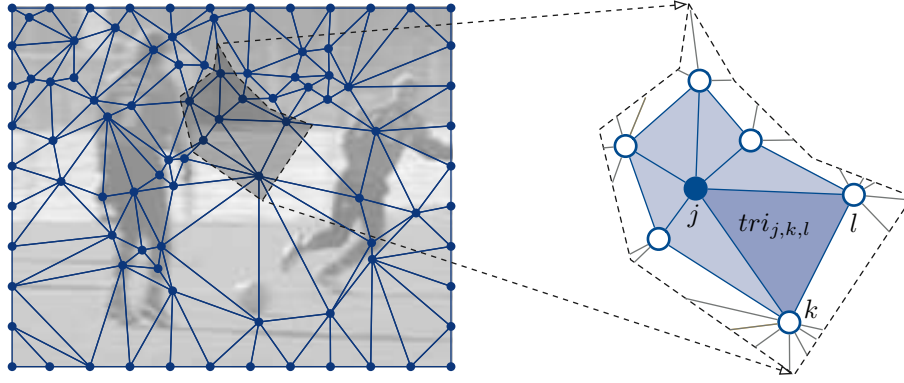


Abbildung 4.14.: Feature-Punkte zur Bestimmung der globalen Bewegung und Wichtung durch umgebende Fläche, **Soccer**, WZ-Frame 76

Merkmalspunkte gespannt wird (Abb. 4.14). Die Delaunay Triangulation wählt die Dreiecke so, dass möglichst kompakte Dreiecke mit großen Innenwinkeln entstehen. Die umgebende Fläche ergibt sich aus der Fläche aller Dreiecke T_j , die den Merkmalspunkt fp_j als Eckpunkt haben (Gl. 4.16).

Die Vektormedianfilterung [Astola u. a. 1990][Alparone u. a. 1996][Ascenso u. a. 2005] wählt den Vektor \mathbf{mv}_i als globalen Bewegungsvektor \mathbf{gmv} aus, welcher den geringsten Abstand zu allen anderen Bewegungsvektoren \mathbf{mv}_j aufweist (Gl. 4.15).

$$\sum_j w_j \|\mathbf{gmv} - \mathbf{mv}_j\|_2 \leq \sum_j w_j \|\mathbf{mv}_i - \mathbf{mv}_j\|_2 \quad (4.15)$$

$$w_j = \sum_{tri_{j,k,l} \in T_j} \text{area}(tri_{j,k,l}) \quad (4.16)$$

$$T_j = \{tri_{s,t,u} | tri_{s,t,u} \in T \wedge j \in \{s, t, u\}\} \quad (4.17)$$

4.4.2. Adaptive Inter-/Extrapolation, unterstützt durch globale Bewegung (GTIE)

Das erste in dieser Arbeit neu entwickelte Verfahren, welches ein Modell für die globale Bewegung verwendet, um die Leistungsfähigkeit der zeitlichen Interpolation zu steigern, ist GTIE (*Global Motion Guided Adaptive Temporal Inter-/Extrapolation*, Abb. 4.16). Dieses Verfahren stellt eine Erweiterung des BiMESS Verfahrens dar.

Im ersten Schritt werden die globalen Bewegungsvektoren $\mathbf{gmv}_{0..2}$ zwischen den jeweils benachbarten Key-Frames ($\mathbf{K}_{i-3} \dots \mathbf{K}_{i+3}$) bestimmt. Nachfolgend wird für jeden Block, basierend auf dem globalen Bewegungsvektor \mathbf{gmv}_1 , der entsprechende Inter-/Extrapolationsmodus festgelegt (Abb. 4.15). Der Modus richtet sich dabei nach der Sichtbarkeit der Bereiche in den Nachbarbildern, wobei 4 Modi unterschieden werden.

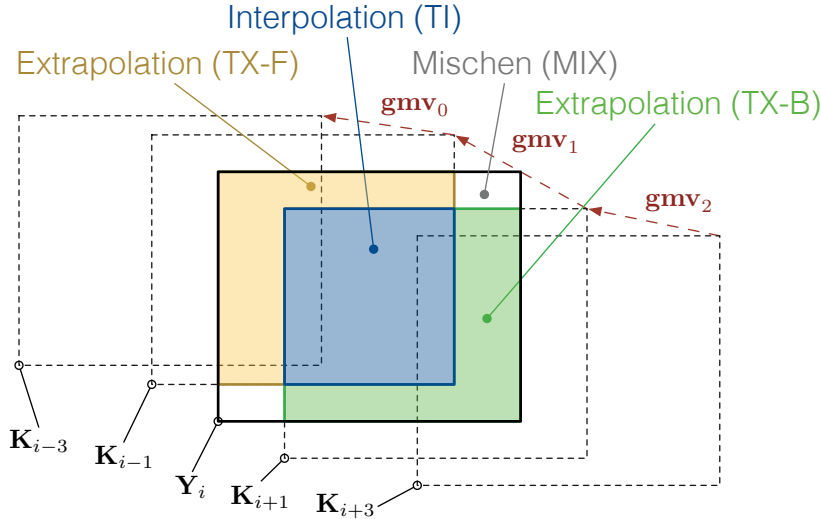


Abbildung 4.15.: Inter- bzw. Extrapolationsmodus im Bildrandbereich in Abhängigkeit vom globalen Bewegungsvektor in GTIE und GpTIE

Interpolation (TI) — Bereiche sind in beiden benachbarten Bildern ($\mathbf{K}_{i-1}, \mathbf{K}_{i+1}$) sichtbar → zeitliche Interpolation (TI), entspricht im Wesentlichen dem Bi-MESS Verfahren ohne örtliche Glättung des Bewegungsvektorfeldes

Vorwärts-Extrapolation (TX-F) — Bereiche sind nur im vorangegangenen Bild \mathbf{K}_{i-1} sichtbar → zeitliche Extrapolation (TX-F) in Vorwärtsrichtung

Rückwärts-Extrapolation (TX-B) — Bereiche sind nur im nachfolgenden Bild \mathbf{K}_{i+1} sichtbar → zeitliche Extrapolation (TX-B) in Rückwärtsrichtung

Mischen (MIX) — Bereiche sind in keinem Bild sichtbar → Mittelung (MIX) über alle anderen Modi

Alternativ zum vierten Modus (Mischen, MIX) ist es hier auch möglich, diese Bereiche durch eine örtliche Extrapolation, wie z.B. *Image Inpainting*, aufzufüllen.

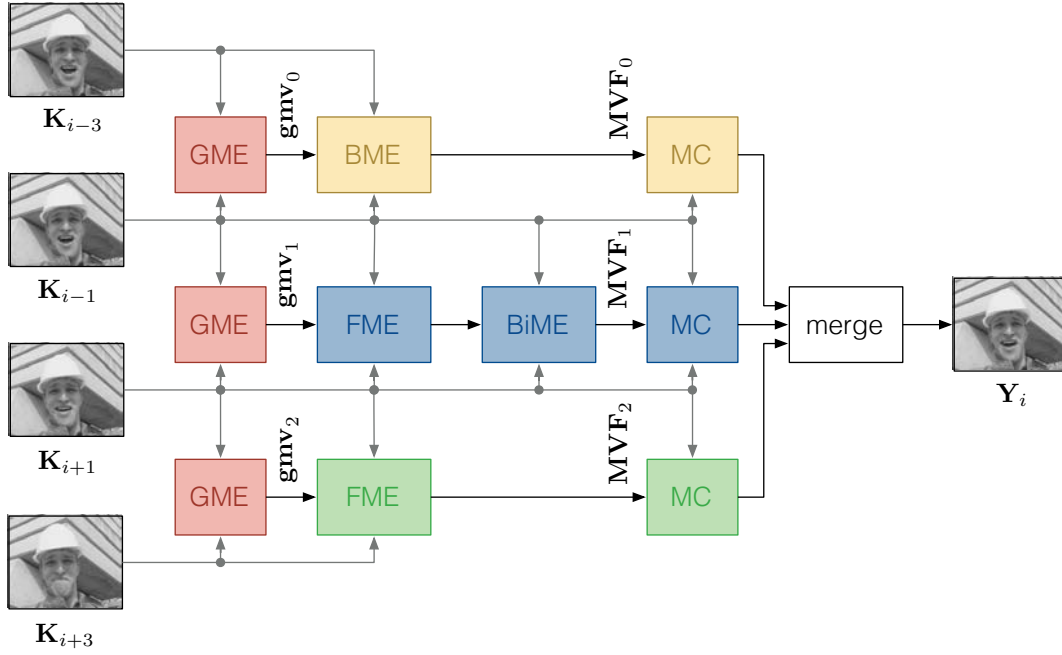


Abbildung 4.16.: Blockschaltbild der geführten temporalen Inter-/Extrapolation (GTIE)

Es hat sich jedoch gezeigt, dass dies nur einen sehr geringen Teil des Bildes betrifft und somit kein wesentlicher Einfluss auf die Qualität der Seiteninformation \mathbf{Y}_i besteht.

Die abschließende Bewegungskompensation und das Erzeugen der Seiteninformation sind detailliert in Gleichung 4.18 dargestellt.

$$\mathbf{Y}_i(x, y) = \begin{cases} \mathbf{K}_{i-1}^{\mathbf{MVF}_0}(x, y) & : mode = \text{TX-F} \\ \frac{1}{2}\mathbf{K}_{i-1}^{\mathbf{MVF}_1}(x, y) + \frac{1}{2}\mathbf{K}_{i+1}^{-\mathbf{MVF}_1}(x, y) & : mode = \text{TI} \\ \mathbf{K}_{i+1}^{-\mathbf{MVF}_2}(x, y) & : mode = \text{TX-B} \\ \frac{1}{4}\mathbf{K}_{i-1}^{\mathbf{MVF}_0}(x, y) + \frac{1}{4}\mathbf{K}_{i-1}^{\mathbf{MVF}_1}(x, y) + \frac{1}{4}\mathbf{K}_{i+1}^{-\mathbf{MVF}_1}(x, y) + \frac{1}{4}\mathbf{K}_{i+1}^{-\mathbf{MVF}_2}(x, y) & : mode = \text{MIX} \end{cases} \quad (4.18)$$

wobei $\mathbf{K}_{i-1}^{\mathbf{MVF}}$ dem bewegungskompensierten Frame \mathbf{K}_{i-1} unter Verwendung des Bewegungsvektorfeldes \mathbf{MVF} entspricht.

Der wesentliche Vorteil des vorgeschlagenen Verfahrens GTIE besteht darin, dass kritische Bereiche, wie Bildrandbereiche durch eine Extrapolation ermittelt werden. Für unkritische Bereiche wird weiterhin die zeitliche Interpolation eingesetzt, da diese hier sehr leistungsfähig ist. Die zeitliche Extrapolation sowie die zeitliche Interpolation verwenden die geschätzten globalen Bewegungsvektoren als Startpunkt. Die Verwendung der zusätzlichen Extrapolation und die globale Bewegungsschätzung erhöhen die Leistungsfähigkeit bei starker Bewegung wesentlich.

Das GTIE Verfahren setzt, wie auch BiMESS, Blockmatching ($8 \times 8 \text{ px}^2$) zur Bewegungsschätzung ein. Da auch die Bewegungskompensation blockbasiert durchgeführt wird, treten in der geschätzten Seiteninformation Blockartefakte auf, was einen der wesentlichen Nachteile von GTIE darstellt.

4.4.3. Pixelbasierte adaptive Inter-/Extrapolation, unterstützt durch globale Bewegung (GpTIE)

Das Problem der Bildung von Blockartefakten wurde in PBTI (Abschn. 4.1) durch den Einsatz einer pixelbasierten Bewegungsschätzung und -kompensation gelöst. Das nun vorgeschlagene Verfahren GpTIE (*Global Motion Guided Pixel-based Adaptive Temporal Inter-/Extrapolation*) erweitert GTIE um den Aspekt der pixelbasierten Bewegungsschätzung und -kompensation.

Die Entscheidung zwischen Inter- bzw. Extrapolation wird, wie bei GTIE, auf Basis des globalen Bewegungsvektors \mathbf{gm}_1 getroffen (Abb. 4.15). Der Interpolationsmodus entspricht dabei weitestgehend dem PBTI Verfahren, welches durch den globalen Bewegungsvektor geführt wird (Abb. 4.17, blau). Die Extrapolationsmodi führen jeweils eine vor-/rückwärts gerichtete pixelbasierte Bewegungsschätzung (pFME, pBME) durch (Abb. 4.17, gelb/grün).

$$\mathbf{Y}'_i(x, y) = \begin{cases} \frac{1}{4}\mathbf{K}_{i-1}^{\frac{1}{2}\mathbf{MVF}_{f1}}(x, y) + \frac{1}{4}\mathbf{K}_{i-1}^{\frac{1}{2}\mathbf{MVF}_{b1}}(x, y) + & : mode = \text{TI} \\ \frac{1}{4}\mathbf{K}_{i+1}^{-\frac{1}{2}\mathbf{MVF}_{f1}}(x, y) + \frac{1}{4}\mathbf{K}_{i+1}^{-\frac{1}{2}\mathbf{MVF}_{b1}}(x, y) & \\ \mathbf{K}_{i-1}^{\frac{1}{2}\mathbf{MVF}_0}(x, y) & : mode = \text{TX-F} \\ \mathbf{K}_{i+1}^{-\frac{1}{2}\mathbf{MVF}_2}(x, y) & : mode = \text{TX-B} \\ \frac{1}{3}\mathbf{K}_{i-1}^{\frac{1}{2}\mathbf{MVF}_0}(x, y) + \frac{1}{12}\mathbf{K}_{i-1}^{\frac{1}{2}\mathbf{MVF}_{f1}}(x, y) + & \\ \frac{1}{12}\mathbf{K}_{i-1}^{\frac{1}{2}\mathbf{MVF}_{b1}}(x, y) + \frac{1}{12}\mathbf{K}_{i+1}^{-\frac{1}{2}\mathbf{MVF}_{f1}}(x, y) + & : mode = \text{MIX} \\ \frac{1}{12}\mathbf{K}_{i+1}^{-\frac{1}{2}\mathbf{MVF}_{b1}}(x, y) + \frac{1}{3}\mathbf{K}_{i+1}^{-\frac{1}{2}\mathbf{MVF}_2}(x, y) & \end{cases} \quad (4.19)$$

Die so erhaltenen Bewegungsvektorfelder werden zur Bewegungskompensation (pMC) des jeweils benachbarten Key-Frames verwendet, wobei die vorläufige Seiteninformation \mathbf{Y}'_i (Gl. 4.19) bestimmt wird. Abschließend wird auch hier, wie bei PBTI, ein Füllen der verbliebenden Lücken (fill) durchgeführt (Abb. 4.17).

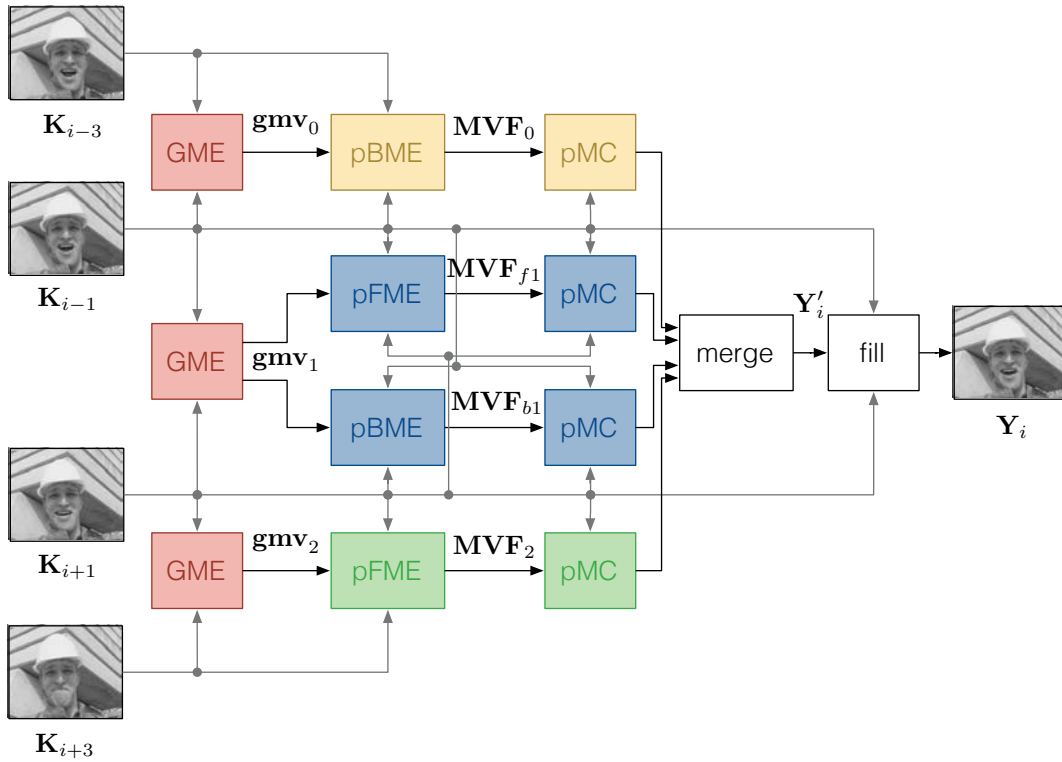


Abbildung 4.17.: Blockschaltbild der pixelbasierten temporalen Inter-/Extrapolation (GpTIE)

4.4.4. Vergleich der entwickelten Verfahren

Die Verfahren GTIE und GpTIE wurden vorgeschlagen, um sich dem Problem der starken globalen Bewegung im Kontext der zeitlichen Interpolation zu widmen. Diese werden nun mit den bekannten Verfahren (BiMESS, PBTI, Abschn. 4.1) bezüglich der objektiven und subjektiven Bildqualität verglichen, um Stärken und Schwächen aufzudecken und um weitere Entwicklungsmöglichkeiten zu finden.

Abbildung 4.18 zeigt einige Beispielbilder der Seiteninformation. Zum einen wurde eine Bildfolge mit geringer Bewegung aus der Sequenz **Foreman** ausgewählt. Aufgrund der geringen globalen Bewegung ist zu erwarten, dass die vorgeschlagenen Verfahren (GTIE, GpTIE) eine ähnliche Leistungsfähigkeit wie die zugrunde liegenden Verfahren (BiMESS, PBTI) zeigen. Das Verfahren GTIE zeigt dem hingegen für das herangezogene Beispielbild eine höhere Qualität als BiMESS. Dieses Verhalten kann durch den Einsatz der örtlichen Glättung des Bewegungsvektorfeldes in BiMESS begründet werden. Hierdurch werden Bewegungsvektoren mit geringen Kosten (SAD) bevorzugt, was vor allem im homogenen Bereich des Helmes negativ hervortritt. Des Weiteren zeigt die pixelbasierte Variante GpTIE eine leichte Verschlechterung bezüglich des PSNR gegenüber PBTI. Dies ist in diesem Bild durch die Ungenauigkeit der Seiteninformationsschätzung am oberen und unteren Bildrand zu begründen, da hier die Extrapolationsmodi (TX-F, TX-B) gewählt wurden.

Zum anderen wurde eine Bildfolge mit starker Bewegung aus der Sequenz **Soccer** ausgewählt (schneller Kameraschwenk nach rechts, siehe Abb. 4.9, S. 51, Bild 152). Hier zeigt sich, dass sowohl GTIE als auch GpTIE einen wesentlichen Vorteil hinsichtlich der Qualität der Seiteninformation liefern. Dies zeigt sich zum einen am objektiven Maß des PSNR. Zum anderen wird der subjektive Bildeindruck verbessert, was an der Hand des Spielers am rechten Bildrand gut zu erkennen ist. Durch den Einsatz der Extrapolationsmodi (TX-F, TX-B) kann auch dieser Bereich am Rande des Bildes gut approximiert werden.

Der Anteil der Pixel in der Seiteninformation, für die Vorwärts- bzw. Rückwärtsextrapolation (TX-F, TX-B) beziehungsweise der Mischmodus (MIX) verwendet wurden, hängt stark von der globalen Bewegung ab und ist mit 1 – 10% nicht sehr hoch (Abb. 4.19, 4.20). Insbesondere ist zu beachten, dass der Modus MIX für Bildbereiche, welche in Nachbarbildern nicht sichtbar sind, sehr selten verwendet wird. Somit



Abbildung 4.18.: Vergleich der Seiteninformationsqualität (GTIE, GpTIE), 30 fps, Foreman WZ-Frame 72, Soccer WZ-Frame 76

hat dieser Modus nur geringen Einfluss auf die Bildqualität. Der Interpolationsmodus wird in allen anderen Fällen verwendet und ist aufgrund der Übersichtlichkeit, in den Abbildungen 4.19 und 4.20, nicht dargestellt.

4.4.5. Fazit

In diesem Abschnitt wurden zwei Verfahren (GTIE, GpTIE) zur Seiteninformationsbestimmung vorgeschlagen, welche die globale Bewegung zur Auswahl des Inter-/Extrapolationsmodus und zur Führung der Bewegungsschätzung verwenden. Die globale Bewegung wird hierbei robust anhand der Bewegung markanter Punkte im Bild geschätzt.

Im Ergebnis der durchgeführten Untersuchungen (Anhang. C, S. 217) konnte ge-

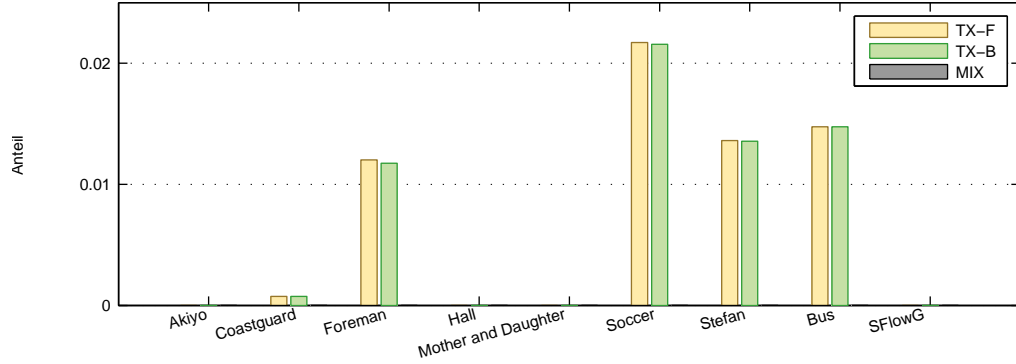


Abbildung 4.19.: Durchschnittlicher Anteil der Bildfläche je Modus, GTIE, Rest Modus TI, 15 fps

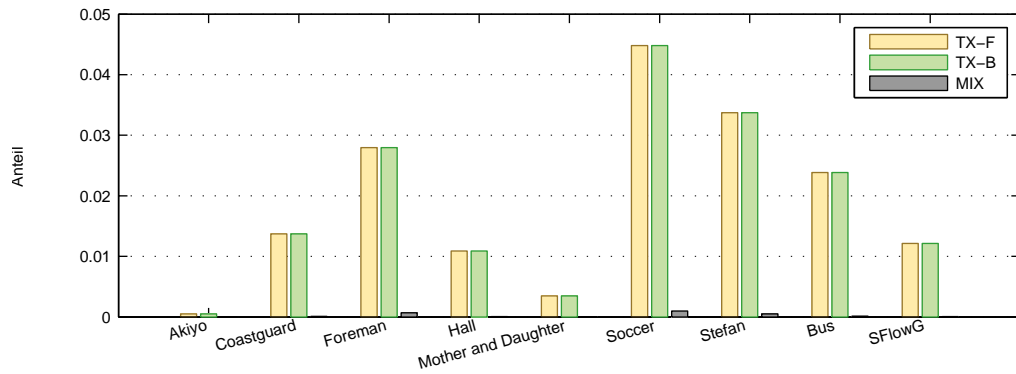


Abbildung 4.20.: Durchschnittlicher Anteil der Bildfläche je Modus, GpTIE, Rest Modus TI, 15 fps

zeigt werden, dass für Sequenzen mit starker Bewegung sowohl eine objektive als auch subjektive Verbesserung der Bildqualität erreicht wird. Weiterhin ist die Performanz für Sequenzen mit geringer Bewegung gleichwertig zu der Leistungsfähigkeit der Basisverfahren (BiMESS, PBTI, basicPBTI).

Die vorgeschlagenen Verfahren (GTIE, GpTIE) setzen eine nahezu lineare Bewegung zwischen den vier benachbarten Key-Frames \mathbf{K}_{i-3} , \mathbf{K}_{i-1} , \mathbf{K}_{i+1} , \mathbf{K}_{i+3} voraus, da eine lineare Inter- und Extrapolation verwendet wird. Weiterhin wird auch vorausgesetzt, dass die globale Bewegung dominiert und es nur geringe Abweichungen von dieser gibt.

4.5. Inhomogene Bewegung

Im vorangegangenen Abschnitt 4.4 wurde das Problem der starken globalen Bewegung betrachtet und eine entsprechende Lösung entwickelt. Starke globale Bewegungen können zum Beispiel, wie beschrieben, durch einen Kameraschwenk hervorgerufen werden. Wird das Augenmerk im Bild nun auf ein bestimmtes schnell bewegtes Objekt gerichtet, welchem die Kamera folgt, so bewegt sich dieses bezüglich der Kamera wenig, wohingegen sich der Hintergrund jedoch sehr stark bewegt. Beispielhaft sei der WZ-Frame in Abbildung 4.21 betrachtet, in dem der Spieler im linken Bildbereich mit dem Kameraschwenk nach rechts läuft. Sowohl der rechte Spieler als auch der Hintergrund bewegen sich somit schnell nach links.

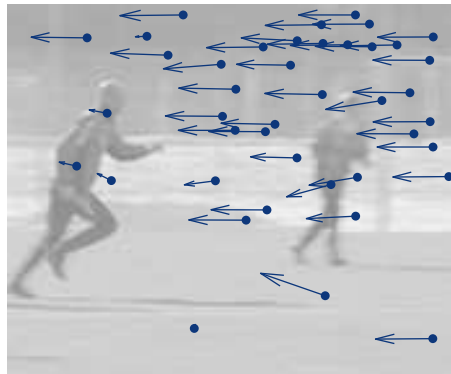


Abbildung 4.21.: Inhomogene Bewegung markanter Punkte, **Soccer**, 30 fps, QCIF, WZ-Frame 64, Referenz: vorangegangener Key-Frame

Die ausschließliche Anwendung der globalen Bewegungsschätzung (Abschn. 4.4), ist für solch ein Szenario aufgrund der inhomogenen Bewegung wenig geeignet. Aus diesem Grund wird hier eine Lösung entwickelt, welche mehrere Hypothesen zur Bewegungsschätzung verwendet (Abb. 4.22). Es werden die folgenden drei Bewegungshypothesen verwendet:

- globale Bewegung
- keine Bewegung
- lokale Bewegung (neu eingeführt)

Das Ziel dieser Herangehensweise besteht in der Sicherstellung, dass die Bewegung jedes Bereiches im Bild mit mindestens einer Hypothese abgedeckt wird.

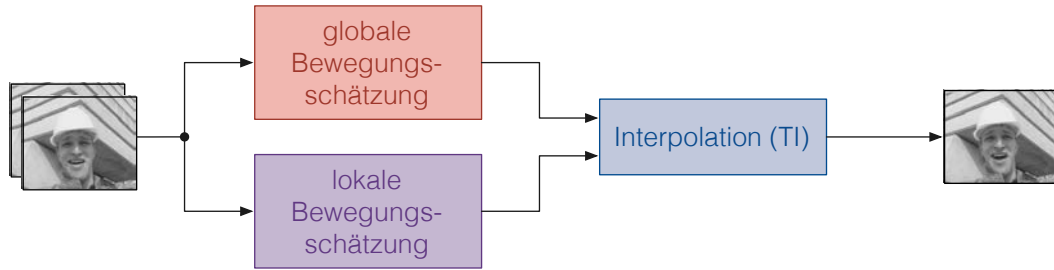


Abbildung 4.22.: Grundstruktur eines Interpolationsverfahrens, basierend auf multiplen Bewegungshypothesen (MPBTI)

In der konventionellen Videocodierung (z.B. H.264 [Wiegand u. a. 2003], HEVC [Bross u. a. 2012]) spielen multiple Hypothesen [Flierl u. a. 1998] bei der bewegungskompensierten Prädiktion eine wichtige Rolle. In diesem Umfeld versteht man unter multiplen Hypothesen die Mittelung über mehrere bewegungskompensierte Bilder (B-Frame). Im Gegensatz dazu werden in dieser Arbeit mit dem Ausdruck „multiple Hypothesen“ multiple Bewegungshypothesen, also die Nutzung mehrerer Startpunkte für die Bewegungssuche, bezeichnet.

Im Folgenden wird die Bestimmung der lokalen Bewegungshypothese dargelegt und darauf aufbauend ein Verfahren zur zeitlichen Interpolation, basierend auf den drei vorgeschlagenen Hypothesen, entwickelt.

4.5.1. Lokale Bewegungshypothese

Die neu eingeführte lokale Bewegungshypothese wie auch die bekannte globale Bewegungshypothese basieren auf der Bewegung markanter Bereiche im Bild. Die Extraktion der Position $(fp_{i,x}, fp_{i,y})$ und Bewegung \mathbf{mv} markanter Bildbereiche fp_i wird, wie in Abschnitt 4.4 vorgeschlagen, auf Basis des SIFT Detektions- und Zuordnungsverfahrens [Lowe 2004] (*feature point detection and matching*) durchgeführt.

Zur Erzeugung der lokalen Bewegungshypothese wird eine Interpolation der Bewegung der benachbarten Feature-Punkte fp_j, fp_k, fp_l angewendet. Dazu wird das Bild mittels Delaunay Triangulation in Dreiecke zerlegt, um die benachbarten Feature-Punkte zu bestimmen. Am Rand des Bildes werden zusätzliche Eckpunkte

eingefügt, um die gesamte Bildfläche abzudecken. Diesen Punkten wird die globale Bewegung zugewiesen (Abb. 4.23).

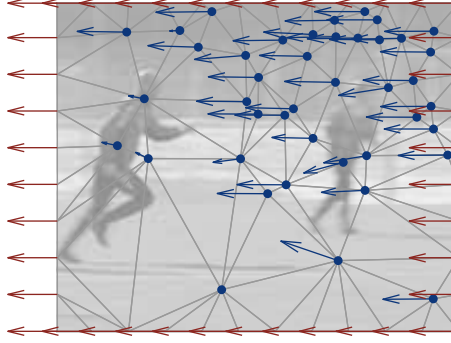


Abbildung 4.23.: Dreiecksnetz und Bewegungsvektoren, basierend auf den Feature-Punkten und zusätzlichen Randpunkten

Innerhalb eines Dreiecks $tri_{j,k,l}$ (Abb. 4.24(a)) wird die lokale Bewegungshypothese \mathbf{hmv}_{local} für den Punkt (x, y) durch Interpolation der Bewegungsvektoren \mathbf{mv}_j der benachbarten Feature-Punkte bestimmt. Dazu wird jeweils für die x- und y-Komponente der Hypothese eine Ebenengleichung (Gl. 4.20) aufgestellt. Die Parameter werden durch Lösen von Gleichung 4.21 bestimmt, womit die Position und Bewegung der Feature-Punkte einbezogen wird. Diese Herangehensweise entspricht einer linearen Interpolation, jeweils getrennt für die x- und y-Komponente.

$$hmv_{local,x} = a_x x + b_x y + c_x \quad (4.20)$$

$$\begin{pmatrix} a_x \\ b_x \\ c_x \end{pmatrix} = \begin{pmatrix} fp_{j,x} & fp_{j,y} & 1 \\ fp_{k,x} & fp_{k,y} & 1 \\ fp_{l,x} & fp_{l,y} & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} mv_{j,x} \\ mv_{k,x} \\ mv_{l,x} \end{pmatrix} \quad (4.21)$$

$$hmv_{local,y} = a_y x + b_y y + c_y \quad (4.22)$$

$$\begin{pmatrix} a_y \\ b_y \\ c_y \end{pmatrix} = \begin{pmatrix} fp_{j,x} & fp_{j,y} & 1 \\ fp_{k,x} & fp_{k,y} & 1 \\ fp_{l,x} & fp_{l,y} & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} mv_{j,y} \\ mv_{k,y} \\ mv_{l,y} \end{pmatrix} \quad (4.23)$$

Exemplarisch ist das interpolierte Hypothesenvektorfeld in Abbildung 4.24(b) dargestellt. Wie zu erkennen ist, passt sich die Hypothese homogen der Bewegung der

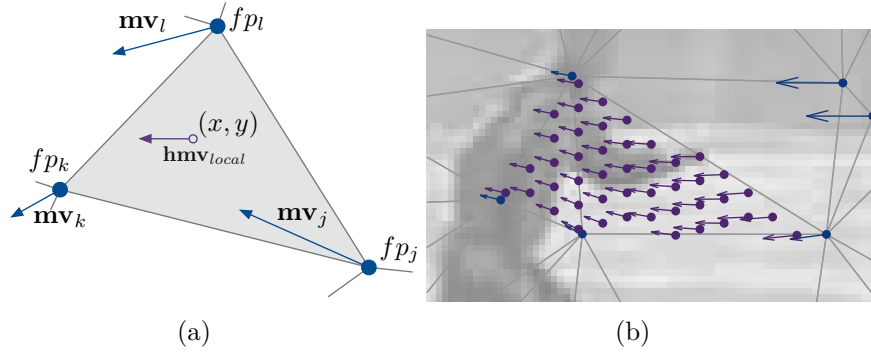


Abbildung 4.24.: Örtliche Interpolation zur Bestimmung der lokalen Bewegungshypothese, (a) verwendetes Dreieck mit Bewegung der Nachbarmerkmalspunkte, (b) interpoliertes dichtetes Bewegungsvektorfeld

Bildbereiche an. Da sich bewegende Segmente aber meist scharfkantig voneinander abgrenzen, kann dies durch die lokale Bewegungshypothese nur bedingt abgebildet werden. Dies muss beim Entwurf des Gesamtverfahrens zur temporalen Interpolation berücksichtigt werden.

Zur Bestimmung der lokalen Bewegungshypothese eines Punktes werden die jeweiligen Nachbarmerkmalspunkte verwendet. Vergleicht man diese Herangehensweise mit der Bestimmung der globalen Bewegungshypothese, so fällt auf, dass hier ein Entfernen von Ausreißern zur Erhöhung der Robustheit nicht vorgenommen wird. Dies lässt erwarten, dass die lokale Hypothese weniger robust ist als die globale Bewegungshypothese.

4.5.2. Multihypothesen pixelbasierte temporale Interpolation (MPBTI)

Basierend auf der vereinfachten pixelbasierten temporalen Interpolation (basicPB-TI, Abschn. 4.3) wird die Verwendung von multiplen Bewegungshypothesen in der Multihypothesen pixelbasierten temporalen Interpolation (MPBTI - *Multiple Motion Hypotheses Pixel-based Temporal Interpolation*) vorgeschlagen. Das Ziel der Anpassung an inhomogene Bewegungen steht dabei im Vordergrund.

Des Weiteren verwendet dieses Verfahren ausschließlich die zwei benachbarten Key-Frames $\mathbf{K}_{i-1}, \mathbf{K}_{i+1}$ zur Generierung der Hypothesen und Bestimmung der Seiten-

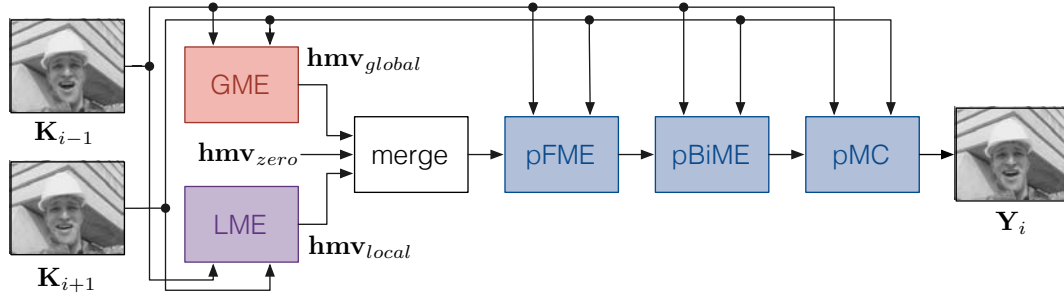


Abbildung 4.25.: Blockschaltbild der Multihypothesen pixelbasierte temporale Interpolation (MPBTI)

information Y_i . Im Gegensatz zu GTIE und GpTIE wird somit keine lineare Bewegung über vier Key-Frames vorausgesetzt, womit jedoch auch die Möglichkeit der temporalen Extrapolation entfällt.

Im ersten Schritt (Abb. 4.25) werden die verschiedenen Bewegungshypothesen erzeugt. Jede Hypothese hat eine spezielle Funktion:

globale Bewegungshypothese hmv_{global} — Erfassen der globalen Kamerabewegung bzw. Hintergrundbewegung

lokale Bewegungshypothese hmv_{local} — Erfassen der Bewegung von Objekten, die sich relativ zum Hintergrund bewegen

Hypothese, dass keine Bewegung stattfindet hmv_{zero} — Erfassen der Bewegung von Objekten, die sich mit der Kamera bewegen oder nur eine sehr geringe Bewegung zeigen – Rückfallmöglichkeit bei Versagen der Hypothesenermittlung

Die Hypothese für „keine Bewegung“ hmv_{zero} stellt sicher, dass bei Versagen der anderen Hypothesen immer mindestens die Leistungsfähigkeit eines nicht hypothesenbasierten Verfahrens erreicht wird.

Nachfolgend werden die Suchbereiche für die einzelnen Hypothesen (Gl. 4.24, Gl. 4.25, hmv_{local} analog zu hmv_{global}) bestimmt und daraus der Gesamtsuchbereich SW_f (Gl. 4.26, Abb. 4.26) durch Vereinigung der Einzelsuchbereiche ermittelt.

$$SW_{f,zero} = \left\{ (mv_x, mv_y) \in \mathbb{Q}_1^2 \mid -SR_f \leq mv_x \leq SR_f \wedge -SR_f \leq mv_y \leq SR_f \right\} \quad (4.24)$$

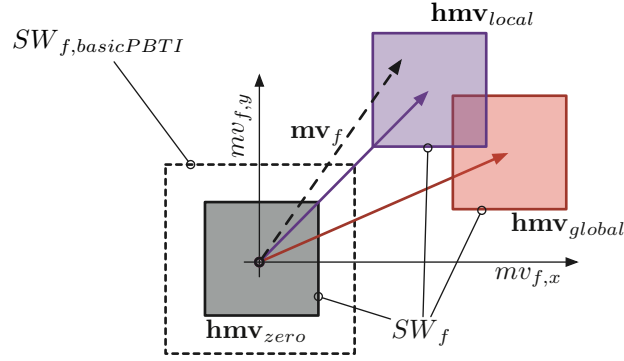


Abbildung 4.26.: Erweiterung des Suchfensters durch multiple Hypothesen

$$SW_{f,global} = \left\{ (mv_x, mv_y) \in \mathbb{Q}_1^2 \mid \begin{aligned} &-SR_f + hmv_{global,x} \leq mv_x \leq SR_f + hmv_{global,x} \wedge \\ &-SR_f + hmv_{global,y} \leq mv_y \leq SR_f + hmv_{global,y} \end{aligned} \right\} \quad (4.25)$$

$$SW_f = SW_{f,zero} \cup SW_{f,global} \cup SW_{f,local} \quad (4.26)$$

Hierbei wird jeweils ein Suchradius SR_f angesetzt (Suchfenster SW_f), welcher kleiner als der im Verfahren basicPBTI ist (Suchfenster $SW_{f,basicPBTI}$), um die Komplexität der Bewegungssuche nicht zu erhöhen. Dieser muss aber auch ausreichend groß gewählt werden, um sprunghafte Änderungen der Bewegung abzudecken, welche durch den homogenen Charakter der lokalen Bewegungshypothese nicht erfasst werden können. Im Fall von QCIF Videosequenzen wird ein Suchradius von $SR_f = 6$ px verwendet, im Vergleich zu basicPBTI mit $SR_f = 10$ px.

Durch die Verschmelzung einzelner rechteckiger Suchbereiche entsteht ein Suchbereich, der nicht zwingend symmetrisch oder rechteckig ist. Dies ist bei starken Bewegungen von Vorteil, da die Fläche des Suchbereichs geringer und somit auch der Suchaufwand reduziert ist. Weiterhin wird, bei zuverlässigen Hypothesen, die Möglichkeit, eine andere als die wahre Bewegung zu detektieren reduziert.

Der weitere Ablauf der Interpolation richtet sich weitestgehend nach der basicPBTI Methode, aus der die pixelbasierte Vorwärtsbewegungsschätzung (pFME), die bidirektionale pixelbasierte Bewegungsschätzung (pBiME) und die abschließende Bewegungskompensation (pMC) übernommen werden.

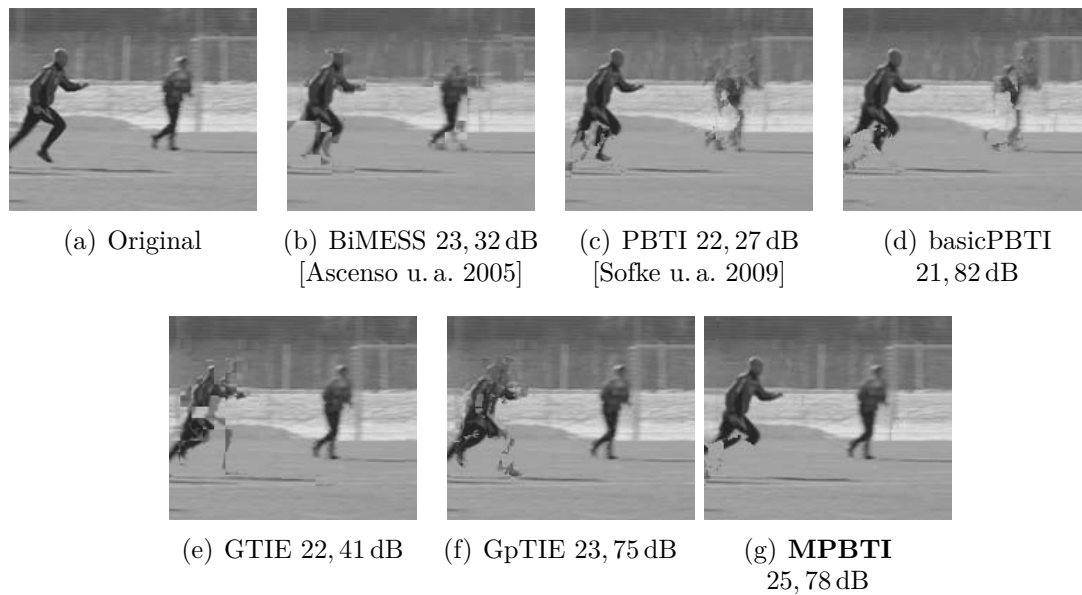


Abbildung 4.27.: Vergleich der Seiteninformationsqualität, Soccer, 30 fps, WZ-Frame 64

4.5.3. Vergleich von MPBTI mit bisherigen Verfahren

In diesem Abschnitt wurde das Problem der inhomogenen Bewegung mit der Entwicklung des Verfahrens MPBTI gelöst. Nun soll das Verfahren MPBTI anhand der Qualität der Seiteninformation mit den bisher bekannten Verfahren verglichen werden. Weiterhin soll bewertet werden, wie groß die Bedeutung der einzelnen Hypothesen ist und inwieweit die Komplexität durch Verwendung der Hypothesen beeinflusst wird.

Qualität der Seiteinformation

Vergleicht man die objektive und visuelle Qualität der Seiteninformation (Abb. 4.27) von MPBTI mit den bisher bekannten Verfahren (BiMESS, PBTI, basicPBTI, GTIE, GpTIE), so ist festzustellen, dass die Leistungsfähigkeit gesteigert werden konnte. Im Vergleich zu BiMESS, PBTI und basicPBTI lässt sich feststellen, dass der Hintergrund sowie der rechte Spieler im Hintergrund aufgrund der sehr starken Bewegung der Kamera nicht gut interpoliert werden konnte. Dieses Problem wurde in GTIE und GpTIE gelöst. Jedoch zeigt sich, dass der Bereich um den linken Spieler im Vordergrund nicht ausreichend gut geschätzt werden konnte, da

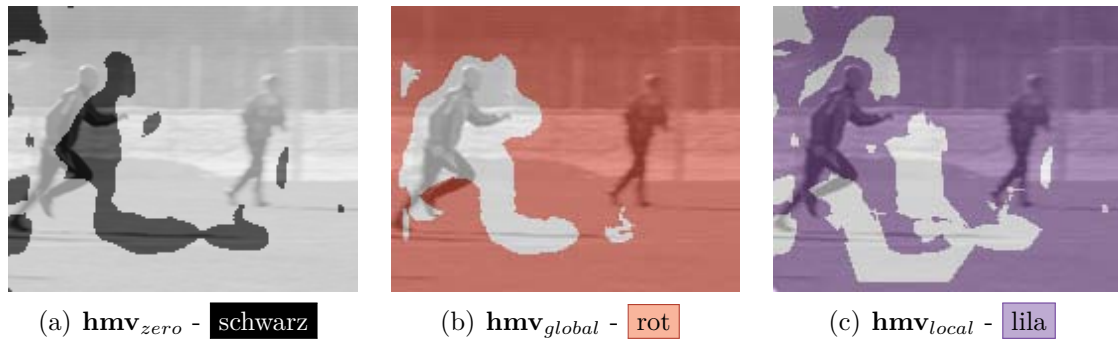


Abbildung 4.28.: Bildbereiche, eingefärbt nach der Hypothese, in deren Suchfenster der Bewegungsvektor liegt, **Soccer**, WZ-Frames 64

dessen Bewegung stark von der globalen Bewegung abweicht. Hier konnte mit Einführung der lokalen Bewegungshypothese sowie der Hypothese „keine Bewegung“ eine wesentliche Verbesserung der objektiven und visuellen Qualität für das MPBTI Verfahren erreicht werden. Eine wesentliche Lücke des MPBTI Verfahrens stellen die fehlenden Extrapolationsmodi (TX-F, TX-B) dar. Dies wird insbesondere am linken Bildrand im Vergleich mit GpTIE deutlich (Abb. 4.27).

Bedeutung der einzelnen Hypothesen

Abbildung 4.28 zeigt die Bereiche im Bild, eingefärbt nach der Hypothese, in deren Suchbereich der Bewegungsvektor liegt. Dabei ist deutlich zu erkennen, dass der Bereich des linken Spielers ausschließlich durch die lokale Bewegungshypothese bestimmt werden kann. Dies begründet die Qualitätsverbesserung für das Beispielbild im Vergleich zu anderen Verfahren (Abb. 4.27). Weiterhin ist festzustellen, dass große Bereiche sowohl durch die globale als auch lokale Bewegungshypothese abgedeckt werden können.

Eine objektive Beurteilung der Bedeutung der einzelnen Hypothesen wird anhand des Anteils durchgeführt, in dem diese These zu einer erfolgreichen Bewegungsschätzung führte (Abb. 4.29). In den Sequenzen mit geringer Bewegung wie **Akiyo**, **Hall** oder **Mother and Daughter** sind die Hypothesen gleichwertig und gegeneinander austauschbar. Hier ist somit die Hypothese „keine Bewegung“ hmv_{zero} ausreichend, was dem Basisverfahren basicPBTI entspricht. Die Sequenzen **Coastguard**, **Foreman**, **Soccer** oder **Stefan** werden durch eine mittlere bis starke homogene

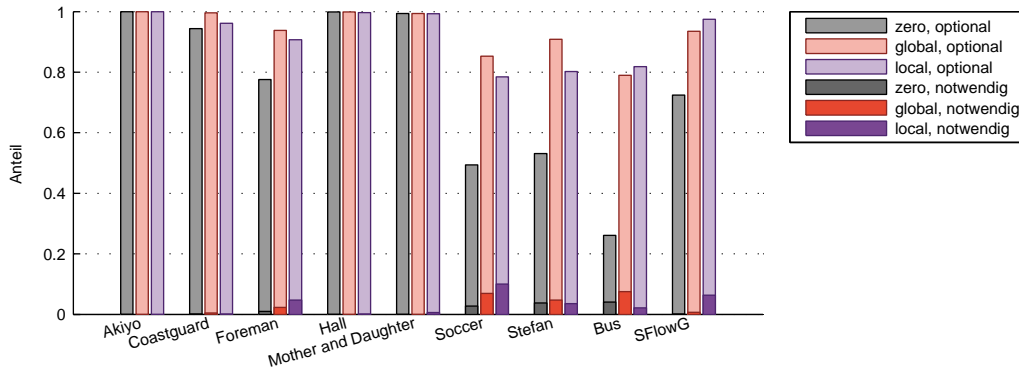


Abbildung 4.29.: Anteil jeder Hypothese, wenn diese optional (hell) oder zwingend zur Bewegungsschätzung notwendig ist (dunkel), 15 fps

globale Bewegung dominiert, da hier die globale Bewegungshypothese \mathbf{hmv}_{global} im Mittel den größten Bildbereich abdeckt. Weiterhin hat die lokale Hypothese \mathbf{hmv}_{local} großes Gewicht in den Sequenzen **Soccer** und **SFlowG**, da diese Hypothese für circa 10 % bzw. 7 % der Pixel notwendig ist und nicht durch eine andere Hypothese ersetzt werden kann.

Größe des Suchfensters

Die Größe des Suchfensters hat wesentlichen Einfluss auf die Komplexität der Bewegungsschätzung. Aus diesem Grund ist in Tabelle 4.1 die mittlere Größe des Suchfensters dargestellt. Im Mittel ist die Suchfenstergröße bei MPBTI kleiner als bei basicPBTI, womit auch die Komplexität für die Vorwärtsbewegungsschätzung geringer ist. Nur in wenigen Fällen einzelner Frames ist diese größer.

4.5.4. Fazit

In diesem Abschnitt wurde das Problem schneller inhomogener Bewegungen durch die Verwendung multipler Hypothesen und einer pixelbasierten Interpolation bearbeitet. Am Ende stand das Interpolationsverfahren MPBTI. Es hat sich gezeigt, dass eine Verbesserung der Bildqualität visuell und objektiv erreicht wird. Weiterhin ist festzustellen, dass die hier verwendeten Hypothesen größtenteils gegeneinander austauschbar und ausschließlich bei sehr starker inhomogener Bewegung alle drei Hypothesen notwendig sind.

	Sequenz	15 fps	30 fps
	Akiyo	171 px ²	170 px ²
	Coastguard	223 px ²	198 px ²
	Foreman	226 px ²	206 px ²
	Hall	173 px ²	173 px ²
	Mother and Daughter	177 px ²	173 px ²
	Soccer	285 px ²	239 px ²
	Stefan	281 px ²	251 px ²
Mittel über alle QCIF Sequenzen		219, 3 px ²	201, 4 px ²
zum Vergleich: basicPBTI (QCIF)		441 px ²	441 px ²

	Sequenz	15 fps	30 fps
	Bus	750 px ²	579 px ²
	SFlowG	495 px ²	434 px ²
Mittel über alle CIF bzw. SIF Sequenzen		634 px ²	513 px ²
zum Vergleich: basicPBTI (CIF/SIF)		961 px ²	961 px ²

Tabelle 4.1.: Mittlere Größe des Suchfensters über alle WZ-Frames einer Sequenz

Weiterhin wurden die Extrapolationsmodi (TX-F, TX-B) aus den Verfahren GTIE und GpTIE (Abschn. 4.4) nicht übernommen, was zu Problemen bei der Erzeugung der Seiteninformation am Bildrand führt (Abb. 4.27).

4.6. Aufdeckung und Verdeckung

Eine zeitliche Interpolation setzt voraus, dass die zu schätzenden Bildbereiche in den zeitlich vorangegangenen und nachfolgenden Bildern sichtbar sind. Dies ist notwendig, da sonst keine valide Bewegungsschätzung möglich ist (Anhang A.2.1, S. 198). Weiterhin treten durch die spätere Mittelung über die bewegungskompensierten Frames Geisterbilder auf, welche durch den Betrachter als sehr störend wahrgenommen werden.

Die Sichtbarkeit von Bildbereichen kann zum Beispiel durch eine starke Veränderung der Beleuchtung sowie durch Auf- und Verdeckungen in Folge von Bewegungen entstehen. Im Abschnitt 4.4 (S. 51) wurde dieses Problem im Rahmen der Kompensation starker Kamerabewegungen schon betrachtet. Hierbei erfolgte eine Beschrän-

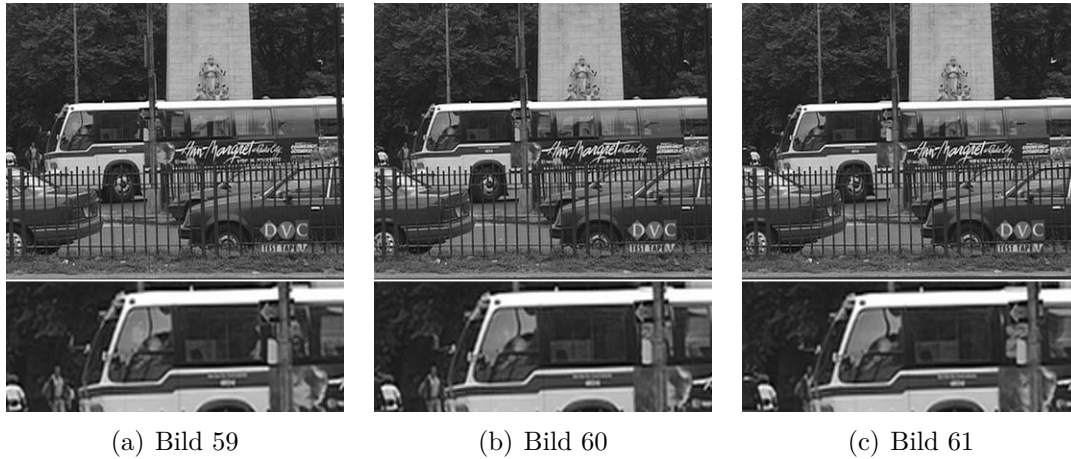


Abbildung 4.30.: Auf- und Verdeckungen an bewegten Objekten, (unten) Detailansicht, Bus, 30 fps

kung auf die am Bildrand auftretenden Auf- und Verdeckungen. Der Schwerpunkt in diesem Abschnitt liegt auf der Behandlung der Auf- und Verdeckungen, welche an bewegten Objekten auftreten.

Ein typischer Fall für Auf- und Verdeckungen ist in Abbildung 4.30 dargestellt. Hier ist in der Detailansicht zu erkennen, dass die Person links vom Bus durch diesen im weiteren Verlauf der Bildfolge verdeckt wird. Das Problem der Seiteninformationsgenerierung soll hier, wie auch schon im Abschnitt 4.4, durch eine zeitliche Vor- bzw. Rückwärtsextrapolation gelöst werden. Darüber hinaus treten zwei wesentliche Probleme auf:

- Detektion der Bereiche zur zeitlichen Extrapolation
- Auswahl des Extrapolationsmodus

Die Lage von Bereichen, in denen eine zeitliche Extrapolation in Frage kommt, muss gesondert bestimmt sowie der Extrapolationsmodus ausgewählt werden. Der Modus (Vor-, Rückwärtsextrapolation) ist dabei davon abhängig, ob eine Ver- bzw. Aufdeckung stattfindet. Die Frage nach der Sichtbarkeit eines Objektes kann durch die relative Tiefe der Objekte beantwortet werden.

Zwei Verfahren (APBTI, APBTI2), welche adaptiv die temporale Extrapolation der temporalen Interpolation vorziehen und den Modus automatisch wählen, werden in diesem Abschnitt entwickelt.

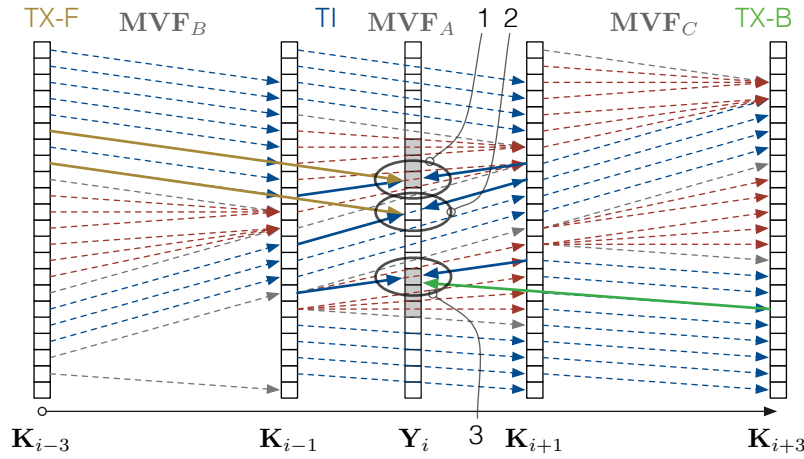


Abbildung 4.31.: Auswahl des Modus (TI, TX-F, TX-B) im Verfahren APBTI, basierend auf den Bewegungsvektorfeldern $\mathbf{MVF}_{A...C}$ und dem SAD der Bewegungsschätzung $\mathbf{SAD}_{A...C}$, die entsprechenden SAD-Werte sind farblich wie folgt gekennzeichnet: rot - hoher, grau - mittlerer, blau - kleiner SAD

4.6.1. Adaptive pixelbasierte zeitliche Interpolation (APBTI)

Die Umschaltung zwischen der direkten zeitlichen Vor-/Rückwärtsextrapolation und Interpolation stellt den Kernpunkt der adaptiven pixelbasierten zeitlichen Interpolation (APBTI) dar. Hier wird, wie auch in GpTIE und MPBTI, die pixelbasierte Bewegungsschätzung und -kompensation verwendet, um mögliche Blockartefakte zu vermeiden. Weiterhin nutzt dieses Verfahren insgesamt vier benachbarte Key-Frames, zwei vor und zwei nach dem aktuellen Bild.

Im ersten Schritt werden zwischen den jeweils benachbarten Key-Frames die Bewegungsvektorfelder $\mathbf{MVF}_{A...C}$ und der zugehörige Matchingfehler $\mathbf{SAD}_{A...C}$ bestimmt (Abb. 4.31), wobei das Bewegungsvektorfeld \mathbf{MVF}_A durch eine Kombination von vorwärts- und rückwärtsgerichteter Bewegungsschätzung erzeugt wird. Die Entscheidung zwischen den drei Modi Interpolation (TI), Vorwärtsextrapolation (TX-F) und Rückwärtsextrapolation (TX-B) wird auf Basis des geringsten SAD durchgeführt (Gl. 4.27).

$$mode_{x,y} = \arg \min(\{sad_{A,x,y}; sad_{B,x,y} + \delta; sad_{C,x,y} + \delta\}) \quad (4.27)$$

Die Extrapolationsmodi erhalten ein zusätzliches Handicap δ , womit eine Bevorzugung der leistungsstarken temporalen Interpolation erreicht wird. Abbildung 4.31 verdeutlicht die drei dabei auftretenden Fälle:

Fall 1 — verdeckter Bereich, welcher durch zeitliche Extrapolation in Vorwärtsrichtung gefüllt werden kann (TX-F)

Fall 2 — sichtbarer Bereich, zeitliche Interpolation, \mathbf{SAD}_B und \mathbf{SAD}_A sind hier gleich, die Interpolation wird aber aufgrund des Handikap δ der Extrapolation vorgezogen (TI)

Fall 3 — aufgedeckter Bereich, rückwärtsgerichtete zeitliche Extrapolation verwenden (TX-B)

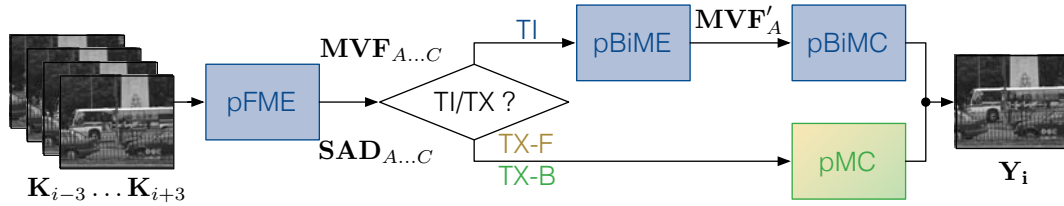


Abbildung 4.32.: Adaptive pixelbasierte zeitliche Interpolation (APBTI)

Nachdem der Modus bestimmt wurde, wird das Bewegungsvektorfeld \mathbf{MVF}_A für die zeitliche Interpolation durch eine bidirektionale Bewegungsschätzung verfeinert (Abb. 4.32). Abschließend wird für jeden Modus eine Bewegungskompensation durchgeführt, um die Seiteninformation \mathbf{Y}_i zu erzeugen.

$$\mathbf{Y}_i(x, y) = \begin{cases} \frac{1}{2}\mathbf{K}_{i-1}^{\frac{1}{2}}\mathbf{MVF}'_A(x, y) + \frac{1}{2}\mathbf{K}_{i+1}^{-\frac{1}{2}}\mathbf{MVF}'_A(x, y) & : mode_{x,y} = \text{TI} \\ \mathbf{K}_{i-1}^{\frac{1}{2}}\mathbf{MVF}_B(x, y) & : mode_{x,y} = \text{TX-F} \\ \mathbf{K}_{i+1}^{-\frac{1}{2}}\mathbf{MVF}_C(x, y) & : mode_{x,y} = \text{TX-B} \end{cases} \quad (4.28)$$

Die Bestimmung der Seiteninformation zu einem Bild beruht hier auf der Ermittlung von vier Bewegungsvektorfeldern ($\mathbf{MVF}_{A...C}$).

Dadurch wird der Rechenaufwand wesentlich erhöht. Dies wird kompensiert, da die gleichen Bewegungsvektorfelder auch für die Bestimmung der Seiteninformationen der benachbarten WZ-Frames notwendig sind, womit die Komplexität pro Frame halbiert wird.

4.6.2. Erweiterte Adaptive pixelbasierte zeitliche Interpolation (APBTI2)

Das zuvor entwickelte Verfahren APBTI wählt zwischen temporaler Interpolation und Extrapolation anhand des SAD aus. Dieses Verfahren soll nun hinsichtlich starker globaler und inhomogener Bewegung erweitert werden. Dazu werden, in der erweiterten Version APBTI2 multiple Bewegungshypothesen verwendet. Weiterhin wird die temporale Extrapolation nicht direkt auf Basis der Bewegungsvektorfelder \mathbf{MVF}_B und \mathbf{MVF}_C durchgeführt, da dies eine lineare Bewegung zwischen den benachbarten vier Key-Frames voraussetzt. In APBTI2 wird alternativ eine örtliche Extrapolation des Bewegungsvektorfeldes \mathbf{MVF}_A angewendet, was ausschließlich Linearität zwischen den beiden benachbarten Key-Frames voraussetzt.

Zur Kompensation von schnellen und inhomogenen Bewegungen wird in APBTI2 (Abb. 4.33) eine Bewegungsschätzung auf Basis multipler Bewegungshypothesen, nach dem Vorbild von MPBTI (Abschn. 4.5.2, S. 68), eingesetzt. Die multiplen Bewegungshypothesen werden ausschließlich genutzt, wenn eine signifikante globale Bewegung ermittelt wurde ($|\mathbf{gmv}| > th_{gmv} = 5 \text{ px}$). Weiterhin werden nur die globale Bewegungshypothese und die Hypothese „keine Bewegung“ aus MPBTI übernommen. Der Grund dafür liegt in der hohen Komplexität der lokalen Hypothese. Weiterhin überlappen die Hypothesen stark (Abb. 4.29 ,S. 73), so kann durch die beiden gewählten Bewegungshypothesen (globale und keine Bewegung) ein Großteil der Ortsveränderungen abgedeckt werden.

Die Entscheidung zwischen den drei Modi (TI, TX-F, TX-B) wird, wie bei APBTI, auf Basis des SAD durchgeführt (Gl. 4.27), jedoch wird ein Extrapolationsmodus ausschließlich dann gewählt, wenn der Pixel vorher als Kandidat für eine

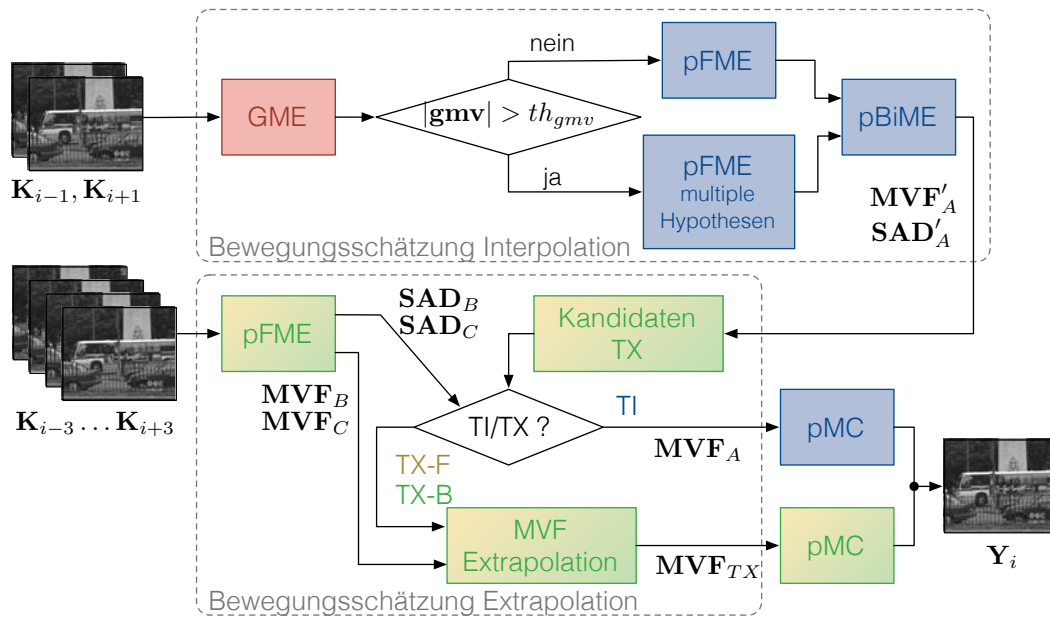


Abbildung 4.33.: Auswahl der potentiellen Pixel zur Nutzung eines der Extrapolationsmodi, basierend auf den Bereichen nicht verbundener Pixel und der ermittelten globalen Bewegung \mathbf{gmv}

Extrapolation ausgewählt wurde. Die Kandidaten werden nach folgenden Kriterien ausgewählt:

globale Bewegung — Randbereiche sind aufgrund der globalen Bewegung \mathbf{gmv} Kandidaten für die Vorwärts- bzw. Rückwärtsextrapolation, je nach Orientierung des globalen Bewegungsvektors (Abb. 4.34)

Diskontinuitäten im Bewegungsvektorfeld \mathbf{MVF}_A — Auf- und Verdeckungen bewirken Diskontinuitäten im Bewegungsvektorfeld \mathbf{MVF}_A . Treten ausreichend große Diskontinuitäten (nicht verbundene Pixel) auf, so werden Pixel innerhalb und am Rand dieser Bereiche als Kandidaten sowohl für die Vor- als auch Rückwärtsextrapolation gekennzeichnet (Abb. 4.34).

Das Bewegungsvektorfeld \mathbf{MVF}_{TX} für die Extrapolation wird nicht direkt aus den Bewegungsvektorfeldern \mathbf{MVF}_B und \mathbf{MVF}_C übernommen, sondern durch örtliche Extrapolation unter Zuhilfenahme der bekannten Bewegungsvektoren ermittelt. Die Bestimmung des Bewegungsvektors \mathbf{mv}_{TX} (Abb. 4.35) bezieht dessen 7×7 -Nachbarschaft (NH) mit ein und beinhaltet folgende Schritte:

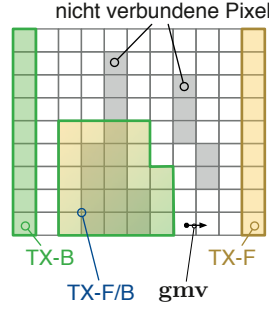


Abbildung 4.34.: Auswahl der potenziellen Pixel zur Nutzung eines der Extrapolationsmodi, basierend auf den Bereichen nicht verbundener Pixel und der ermittelten globalen Bewegung \mathbf{gmv}

1. Prüfe, ob die Nachbarschaft NH mindestens 9 bekannte Bewegungsvektoren enthält.
2. Bestimme den Bewegungsvektor \mathbf{mv}_{TX} nach Gleichung 4.29.
3. Wenn kein Pixel mehr bearbeitet werden kann, füge die ermittelten Bewegungsvektoren \mathbf{mv}_{TX} der Liste der bekannten Vektoren hinzu.
4. Wiederhole, bis alle Bewegungsvektoren bestimmt sind.

$$\begin{aligned} \mathbf{mv}_{TX} &= \arg \min_{\mathbf{mv}_i} \{d(\mathbf{mv}_B, \mathbf{mv}_i) + d(\overline{\mathbf{mv}_{NH}}, \mathbf{mv}_i)\} \\ \mathbf{mv}_B &= \mathbf{MVF}_B(x, y), \quad \mathbf{mv}_i \in NH \end{aligned} \quad (4.29)$$

Gleichung 4.29 gilt für die temporale Vorwärtsextrapolation. Für die Rückwärtsextrapolation wird \mathbf{MVF}_B durch \mathbf{MVF}_C ersetzt. Weiterhin gibt $d(\dots)$ die euklidische Distanz und $\overline{\mathbf{mv}_{NH}}$ die mittlere Bewegung der Nachbarschaft an.

Die örtliche Extrapolation des Bewegungsvektorfeldes (Gl. 4.29) basiert auf der Idee, einen Bewegungsvektor aus der Nachbarschaft zu wählen, welcher dem Bewegungsvektor \mathbf{mv}_B bzw. \mathbf{mv}_C (Bewegung zwischen den äußeren Key-Frames) möglichst nahekommt, jedoch von der mittleren Bewegung der Nachbarschaft nicht stark abweicht. Hierbei wird die starke örtliche Korrelation des Bewegungsvektorfeldes ausgenutzt. Die Bewegungsvektoren zwischen den benachbarten Key-Frames werden nicht direkt übernommen, da dies eine lineare Bewegung zwischen allen vier Key-Frames voraussetzt. Durch die Bestimmung eines ähnlichen Bewegungsvektors



kann das Problem, nichtlineare Bewegung zwischen den vier benachbarten Key-Frames, umgangen werden. Die Linearität der Bewegung zwischen den benachbarten Key-Frames \mathbf{K}_{i-1} , \mathbf{K}_{i+1} bleibt weiterhin eine Voraussetzung.

$$\mathbf{Y}_{: (x, y)} = \begin{cases} \frac{1}{2} \mathbf{K}_{i-1}^{\frac{1}{2} \mathbf{MVF}'_A}(x, y) + \frac{1}{2} \mathbf{K}_{i+1}^{-\frac{1}{2} \mathbf{MVF}'_A}(x, y) & : mode_{x,y} = \text{TI} \\ \mathbf{K}_{i-1}^{\frac{1}{2} \mathbf{MVF}_{TX}}(x, y) & : mode_{x,y} = \text{TX-F} \end{cases} \quad (4.30)$$

$$\mathbf{Y}_i(x, y) = \begin{cases} \frac{1}{2}\mathbf{K}_{i-1}^{\frac{1}{2}\mathbf{MVF}'^A}(x, y) + \frac{1}{2}\mathbf{K}_{i+1}^{-\frac{1}{2}\mathbf{MVF}'^A}(x, y) & : mode_{x,y} = \text{TI} \\ \mathbf{K}_{i-1}^{\frac{1}{2}\mathbf{MVF}^{TX}}(x, y) & : mode_{x,y} = \text{TX-F} \\ \mathbf{K}_{i+1}^{-\frac{1}{2}\mathbf{MVF}^{TX}}(x, y) & : mode_{x,y} = \text{TX-B} \end{cases} \quad (4.30)$$

Vergleicht man die subjektive und objektive Qualität der Seiteninformation von APBTI und APBTI2 mit den bisher entwickelten und aus der Literatur bekannten Verfahren (Abb. 4.36), so ist zu erkennen, dass die Qualität leicht verbessert werden kann. Subjektive Verbesserungen sind besonders in der Detailansicht (Abb. 4.36) links vom Bus zu erkennen, da hier, je nach verwendeten Verfahren,

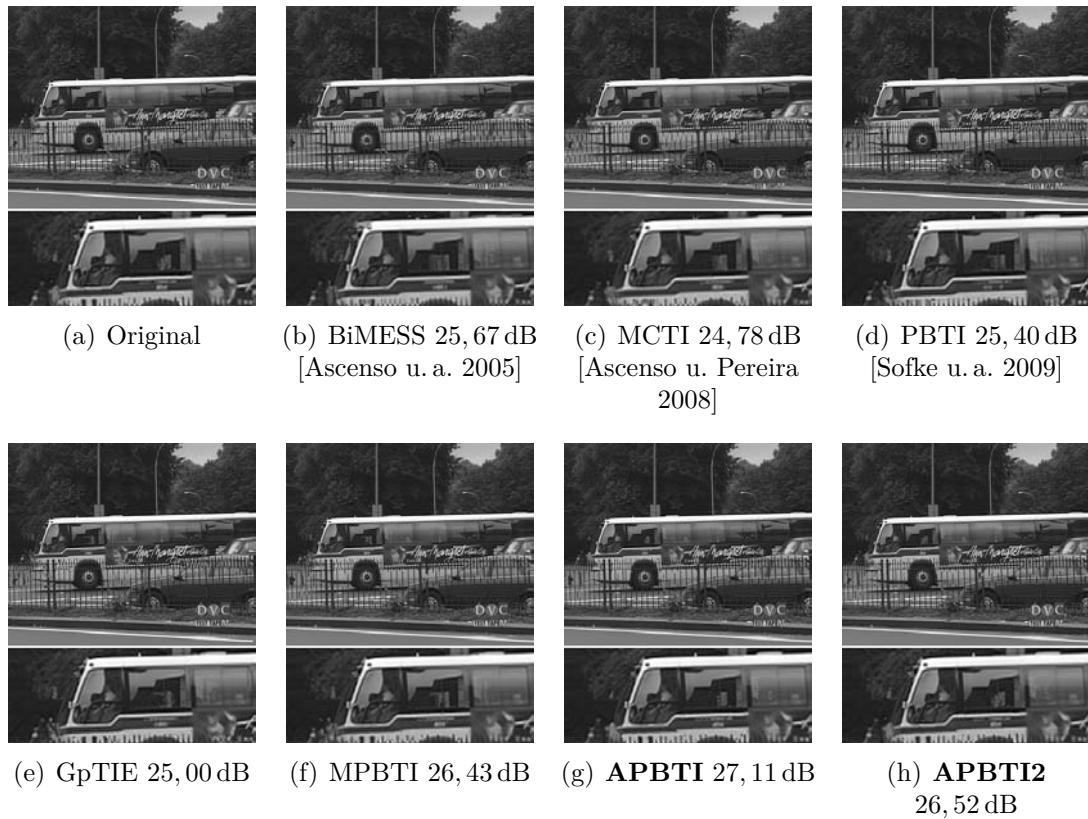


Abbildung 4.36.: Vergleich von APBTI(2) mit bekannten Verfahren anhand der Seiteninformationsqualität, *Bus*, 30 fps, WZ-Frame 30, PSNR für das gesamte Bild

starke Artefakte auftreten. Bei diesem Bereich handelt es sich um einen Verdeckungsbereich, in dem sich der Bus vor dem Hintergrund von rechts nach links bewegt.

4.6.4. Fazit

In diesem Abschnitt wurde das Problem von Auf- und Verdeckungen in der Seiteninformationsgenerierung mittels zeitlicher Interpolation betrachtet. Dazu wurden zwei Verfahren (APBTI, APBTI2) entwickelt, welche eine adaptive pixelbasierte zeitliche Interpolation einsetzen. Die Idee liegt darin, in Auf- und Verdeckungsbe-
reichen eine temporale Extrapolation zu verwenden. In APBTI werden die Bewegungsvektoren für die Extrapolation auf Basis der vier benachbarten Key-Frames bestimmt, was eine lineare Bewegung voraussetzt. Um dieses Problem zu umgehen,

wurde in APBTI2 eine örtliche Extrapolation des Bewegungsvektorfeldes eingesetzt.

Die Verbesserung der objektiven Qualität der Seiteninformation ist gering. Visuell störende Artefakte, die die subjektive Qualität beeinflussen, konnten jedoch reduziert werden.

4.7. Komplexität

Neben der Encoderkomplexität spielt der Rechenzeitbedarf des Decoders auch eine wichtige Rolle. Dieser ist durch Randbedingungen, wie verfügbare Rechenleistung oder Energie, nicht so stark beschränkt wie der Encoder, jedoch muss auch hier die Komplexität im technisch machbaren Rahmen bleiben.

Einen wesentlichen Beitrag zur Komplexität der Decoders liefert die Schätzung der Seiteninformation. Aus diesen Grund soll das Verfahren basicPBTI (Abschn. 4.3, S. 47) bewertet und mit anderen Verfahren bezüglich seiner Komplexität und Leistungsfähigkeit verglichen werden. Weiterhin sollen Methoden vorgestellt werden, welche eine Reduktion der Komplexität ermöglichen, die Leistungsfähigkeit aber nur wenig beeinflussen. Ein wesentliches Augenmerk soll auf dem dichten Bewegungsvektorfeld liegen, da dies wesentlich zur Leistungsfähigkeit beiträgt.

4.7.1. Einflussfaktoren auf die Komplexität

Die Bestimmung eines dichten Bewegungsvektorfeldes in basicPBTI wird auf Basis des modifizierten Blockmatchings durchgeführt. Es wird ein gewichteter SAD (wSAD, *weighted sum of absolute difference*) als Distanzmaß verwendet.

Betrachtet man die Komplexität der Vorwärtsbewegungsschätzung, so wird die Anzahl der notwendigen arithmetischen Operationen im Wesentlichen durch drei Faktoren beeinflusst (Abb. 4.37, Gl. 4.31):

MVF — Mächtigkeit des Bewegungsvektorfeldes (Anzahl der zu bestimmenden Bewegungsvektoren)

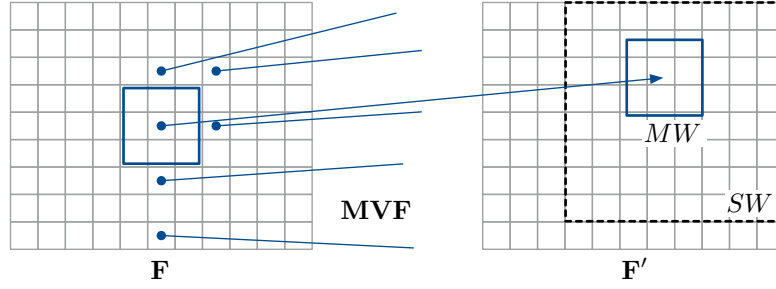


Abbildung 4.37.: Einflussfaktoren auf die Komplexität der Bewegungsschätzung auf Basis von Blockmatchingverfahren

SW — die Größe des Suchbereiches (Anzahl der möglichen Werte eines Bewegungsvektors)

MW — die Größe des Matching-Fensters

$$\begin{aligned}
 \mathbf{mv}_f &= \arg \min_{\mathbf{mv}_u} \sum_{(k,l) \in MW} \mathbf{GW}_\sigma(k,l) \cdot \left| \mathbf{F}(x+k, y+l) - \right. \\
 &\quad \left. \mathbf{F}'(x+k+mv_{f,x}, y+l+mv_{f,y}) \right| \\
 &\quad \mathbf{mv}_u \in SW
 \end{aligned} \tag{4.31}$$

Für ein Blockmatching mit gegebener Dichte des Bewegungsvektorfeldes \mathbf{MVF} , Größe des Suchfensters SW und gegebener Größe des Matchingfensters MW lässt sich die Anzahl der arithmetischen Operationen nach den Gleichungen 4.32, 4.33 und 4.34 bestimmen. Im Fall eines einfachen Blockmatchings mittels SAD (*sum of absolute difference*) sind nur $N_{Addition}$ Additionen und N_{Betrag} Betragsbildungen notwendig. Das Verfahren basicPBTi setzt einen gewichteten SAD (wSAD) ein, wodurch zusätzlich $N_{Multiplikation}$ Multiplikationen erforderlich sind.

$$\begin{aligned}
 N_{Addition} &= |\mathbf{MVF}| (2 \cdot |SW| \cdot |MW| - 1) \\
 &\approx 2 \cdot |\mathbf{MVF}| \cdot |SW| \cdot |MW|
 \end{aligned} \tag{4.32}$$

$$N_{Betrag} = |\mathbf{MVF}| \cdot |SW| \cdot |MW| \tag{4.33}$$

$$N_{Multiplikation} = |\mathbf{MVF}| \cdot |SW| \cdot |MW| \tag{4.34}$$

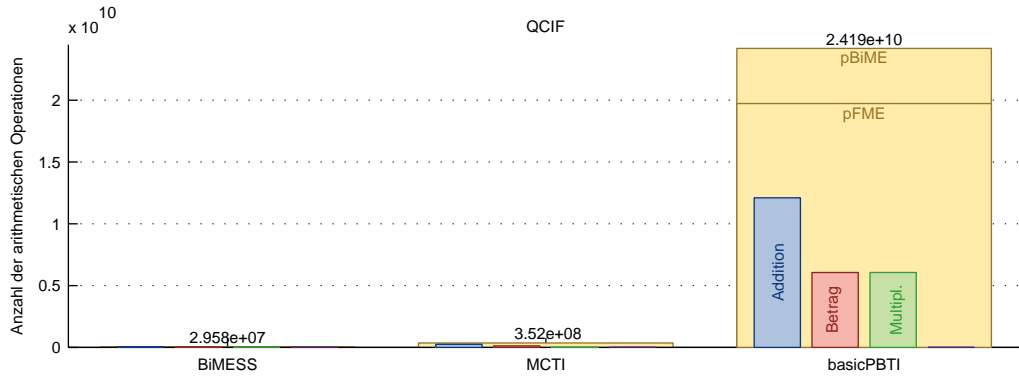


Abbildung 4.38.: Anzahl der arithmetischen Operationen in Verfahren zur zeitlichen Interpolation auf Basis von Blockmatching

Die hier dargestellte Anzahl arithmetischer Operationen für die Vorwärtsbewegungsschätzung (FME) ist auch für eine bidirektionale Bewegungsschätzung (BiME) gültig. Die bidirektionale Bewegungsschätzung wird in basicPBTI mit Halb-Pixel Genauigkeit eingesetzt. Dies erhöht die Anzahl der arithmetischen Operationen um den Faktor 16, da zum einen die Größe des Suchbereiches SW und zum anderen die Größe des Matchingfensters je räumlicher Dimension verdoppelt wird.

Vergleicht man die Anzahl der arithmetischen Operationen, welche für die Generierung der Seiteninformation mittels BiMESS, MCTI oder basicPBTI (Abb. 4.38) aufgewendet werden müssen, so fällt auf, dass für basicPBTI circa 70 mal mehr Operationen notwendig sind als für MCTI.

Dabei entfällt ein Viertel der Operationen auf Multiplikationen, welche je nach Rechenwerk einen größeren Aufwand als Additionen verursachen. Weiterhin wird die größte Rechenzeit durch die Verarbeitungsstufe zur pixelbasierten Vorwärtsbewegungsschätzung (pFME) aufgewendet.

Im Folgenden sollen Ansätze diskutiert werden, welche es ermöglichen, die Komplexität wesentlich zu verringern und so möglichst ein Komplexitätsniveau anderer vergleichbarer Verfahren (BiMESS, MCTI) zu erreichen.

*Verringerung der Mächtigkeit des Bewegungsvektorfeldes **MVF*** — Dies ist einfach möglich, indem zum Beispiel nur für jeden 4-ten Pixel ein Bewegungsvektor bestimmt wird, jedoch muss eine Möglichkeit gefunden werden, weiterhin ein

dichtes Bewegungsvektorfeld zu ermitteln. Weiterhin kann sich die Schätzung eines ausgedünnten Bewegungsvektorfeldes auch auf geeignete einzigartige Bildbereiche, mit zum Beispiel starken Gradienten, stützen.

Verkleinerung des Matching-Fensters MW — Dies ist nicht ohne Weiteres möglich, da die Robustheit der Bewegungsschätzung bei kleinerem Matchingfenster stark eingeschränkt sein kann.

keine Wichtung des SAD durchführen — Die Wichtung der Nachbarschaft des Verfahrens basicPBTI sowie des Ursprungsverfahrens PBTI [Sofke u. a. 2009] stellt eine wesentliche Komponente dar, welche Robustheit und Detailtreue verknüpft. Aus diesem Grund soll das Element der Wichtung erhalten bleiben.

Verkleinerung des Suchbereiches SW — Ein ausreichend großer Suchbereich ist Voraussetzung zur Erfassung starker/inhomogener Bewegungen. Mit ± 10 px ist dieser bei basicPBTI nicht weiter zu verkleinern.

Suchgenauigkeit von Halb- auf Voll-Pixel verringern — Da ausschließlich die pixelbasierte bidirektionale Bewegungsschätzung mit Halb-Pixel Genauigkeit arbeitet, würde so nur eine geringe Reduktion der Rechenzeit erreicht werden.

4.7.2. Ganzzahlige Koeffizienten zur Bestimmung des gewichteten SAD

Die Wichtung des SAD mit einem Gauß-Fenster \mathbf{GW}_σ erhöht die Komplexität des Blockmatchings stark, da so eine große Anzahl Multiplikationen durchgeführt werden müssen (Gl. 4.34, Abb. 4.38). Moderne Rechenwerke arbeiten zumeist mit dem binären Zahlensystem, in welchem sich Multiplikationen mit Faktoren 2^α , $\alpha \in \mathbb{Z}$, sehr einfach durch bitweise Verschiebung realisieren lassen.

Das Ziel besteht nun darin, eine Approximation \mathbf{GW}'_σ des Gauß-Fensters \mathbf{GW}_σ zu finden, welche ausschließlich 2-er Potenzen enthält. Weiterhin wird die Überlegung zugrunde gelegt, dass eine Skalierung aller Elemente des Fensters keinen Einfluss auf das Blockmatching hat, da hier die wSAD Werte ausschließlich miteinander verglichen werden.

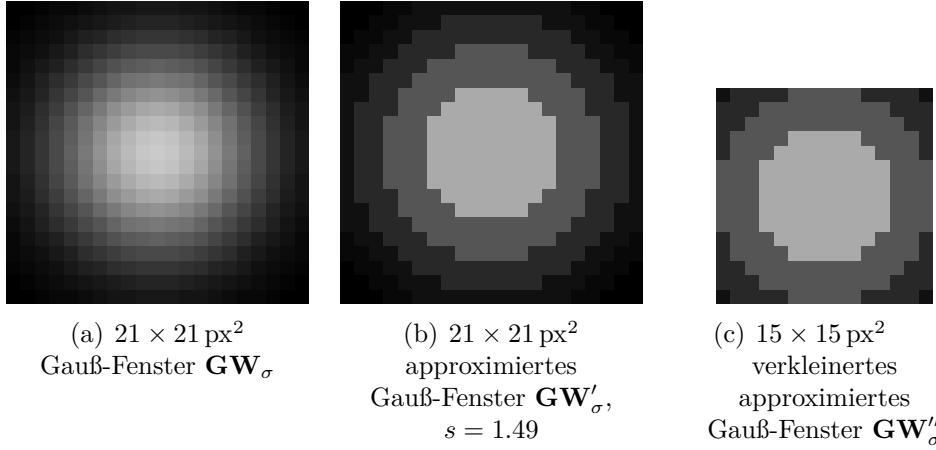


Abbildung 4.39.: Fenster zur Berechnung des wSAD in der pixelbasierten Bewegungsschätzung (pFME, pBiME)

Die beste Approximation des Gauß-Fensters und die beste Skalierung s stellt das skalierte Fenster $s \cdot \mathbf{GW}'_\sigma$ dar, welches den quadratischen Fehler zum Gauß-Fenster \mathbf{GW}_σ minimiert (Gl. 4.35).

$$\mathbf{GW}'_\sigma(k, l) = \arg \min_{\mathbf{GW}'_{\sigma, s}} \sum_{k, l} \left\{ \mathbf{GW}_\sigma(k, l) - s \cdot \mathbf{GW}'_\sigma(k, l) \right\}^2 \quad (4.35)$$

$$\mathbf{GW}'_\sigma(k, l) = 2^\alpha, \quad \alpha \in \mathbb{Z}, \quad s \in [1; 2[$$

Für eine gegebene Skalierung s lässt sich das approximierte Gauß-Fenster \mathbf{GW}'_σ finden, indem jeder Wert $\mathbf{GW}_\sigma(k, l)$ zum nächstgelegenen Wert 2^α gerundet wird. Zur Bestimmung der Skalierung s kann ein Gradientenabstiegsverfahren oder auch eine erschöpfende Suche verwendet werden.

Abbildung 4.39 zeigt das Gauß-Fenster \mathbf{GW}_σ und dessen Approximation \mathbf{GW}'_σ , der Fehler wird bei einer Skalierung mit $s = 1.49$ minimiert. Das approximierte Fenster der Größe $21 \times 21 \text{ px}^2$ enthält viele sehr kleine Gewichte. Aus diesem Grund wird ein auf die Größe $15 \times 15 \text{ px}^2$ verkleinertes Fenster \mathbf{GW}''_σ verwendet, welches ausschließlich die signifikanten Gewichte enthält.

Durch die vorgeschlagene Approximation und Verkleinerung des Fensters werden zum einen die notwendigen Multiplikationen durch wesentlich einfacher umzusetzende Bitverschiebungen realisiert. Weiterhin kann durch Verkleinerung des Fens-

ters die Anzahl der arithmetischen Operationen von $21 \cdot 21 = 441$ auf $15 \cdot 15 = 225$ je Blockvergleich reduziert werden.

4.7.3. Reduktion der Dichte des Bewegungsvektorfeldes (Vorwärtsbewegungsschätzung)

Die Reduzierung der Mächtigkeit des Bewegungsvektorfeldes **MVF** bietet das größte Potenzial zur Reduktion der Komplexität von basicPBTI. Wie jedoch anfangs als Zielsetzung formuliert wurde, sollen die Vorteile durch ein dichtes Bewegungsvektorfeld weiterhin erhalten bleiben. Somit wird vorerst nur die Ausdünnung des Bewegungsvektorfeldes **MVF_f** zur pixelbasierten Vorwärtsbewegungsschätzung betrachtet.

Die Zuverlässigkeit der Bewegungsschätzung hängt stark von der Charakteristik der zur Schätzung verwendeten Nachbarschaft ab. Die Bewegungsschätzung in homogenen Bildbereichen liefert meist unzuverlässige Ergebnisse. Die Nachbarschaft sollte signifikante Gradienten orthogonal zueinander enthalten, um eine zuverlässige Bewegungsschätzung zu ermöglichen (siehe Aperturbedingung, Anhang S. 204). Aus diesem Grund werden die Positionen der Vektoren für die Vorwärtsbewegungsschätzungen nicht einfach in einem Raster angeordnet, sondern anhand der Harris Detektors (Anhang A.2.2, S. 207, [Harris u. Stephens 1988]) ausgewählt.

Der zur Auswahl verwendete Algorithmus (Abb. 4.40) besteht aus vier Stufen, wobei beginnend für alle auf einem $4 \times 4 \text{ px}^2$ Raster liegenden Punkte die Harris-Cornerness $\mathbf{R}_{\text{cornerness}}$ (mit $k = 0.04$) und die Spur (*Trace*) \mathbf{Tr} der Hesse-Matrix bestimmt wird. Die Harris-Cornerness $\mathbf{R}_{\text{cornerness}}$ gibt an, ob es sich um eine Ecke oder Kante handelt, wobei mittels der Spur schwache Ecken detektiert werden können. Die Auswahl der Startpunkte erfolgt nach den folgenden Kriterien:

markante Punkte — Auswahl von Punkten (u, v) , die über eine Spur $\mathbf{Tr}(u, v) \geq 0.005$ und einer hohen Cornerness $\mathbf{R}_{\text{cornerness}}(u, v)$ verfügen. Diese Punkte dürfen nicht im Abdeckungsbereich anderer Punkte liegen (Abb. 4.40, grau), um starke Häufungen in strukturierten Bereichen zu vermeiden.

Abdeckung der Bildfläche — Kein Pixel auf dem Bild soll einen zu großen Abstand vom nächsten markanten Punkt und somit Bewegungsvektor haben.

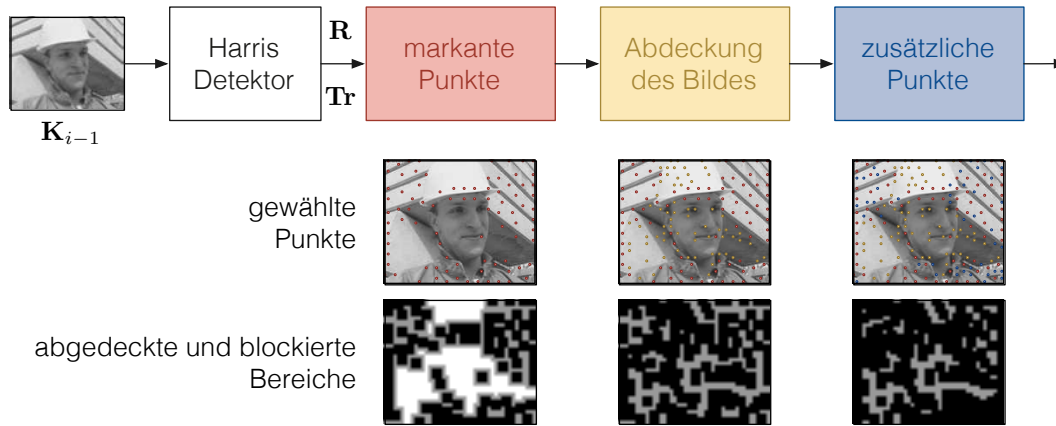


Abbildung 4.40.: Auswahl markanter Bereiche für die pixelbasierte Vorwärtsbewegungsschätzung, Foreman, 30 fps, WZ-Frame, abgedeckter grau und blockierter Bereich schwarz

Für einen Punkt, der nicht im Abdeckungsbereich liegt, wird der Nachbar mit der höchsten Cornerness $R_{cornerness}(u, v)$ gewählt, welcher diesen abdecken kann. Punkte mit einer Spur $Tr(u, v) \geq 0.005$ werden bevorzugt. Der neu hinzugefügte Punkt darf nicht in einem blockierten Bereich liegen, um einen Mindestabstand zwischen den Bewegungsvektoren zu gewährleisten.

zusätzliche Punkte — Weitere charakteristische Punkte werden hinzugefügt, bis eine maximale Anzahl von Punkten erreicht ist. Diese Punkte können in abgedeckten, aber nicht in blockierten Bereich platziert werden.

Der Abdeckungs- und Blockierungsbereich beschreibt den Bildbereich, welcher einen geringeren Abstand als 1 bzw. 2 Rasterschritte von schon gewählten Punkten besitzt (siehe Abb. 4.40, grau bzw. schwarz).

Durch diese Herangehensweise bei der Auswahl von Punkten, beginnend mit den markantesten, ist sichergestellt, dass ausschließlich die aufgrund ihrer Umgebung gut zur Bewegungsschätzung geeigneten Punkte verwendet werden. Weiterhin wird eine Minstdichte (maximaler Abstand) der ausgewählten Punkte sichergestellt. Abbildung 4.41 zeigt die in jedem Schritt gewählten markanten Punkte.

Eine Komplexitätsreduktion für die Vorwärtsbewegungsschätzung wird hier durch Ausdünnung des Bewegungsvektorfeldes MVF_f erreicht. Pro 128 Pixel wird nur ein Bewegungsvektor ausgewählt, was einer Reduktion der Komplexität um den Faktor $\frac{1}{128}$ entspricht.

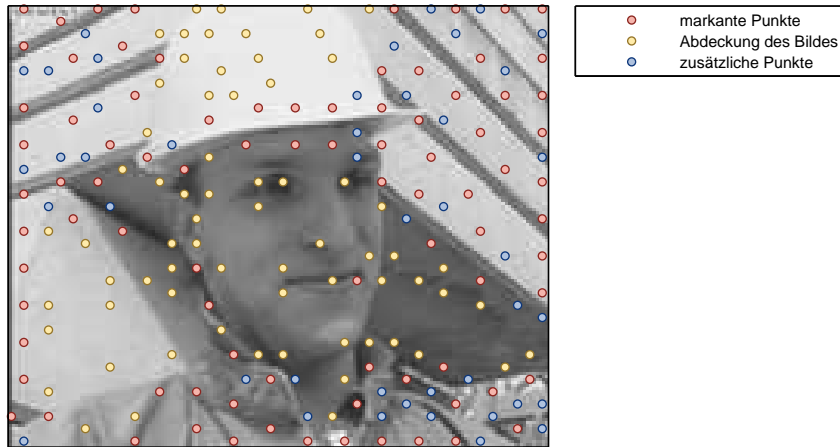


Abbildung 4.41.: Ausgewählte markante Positionen (hohe Cornerness) für die pixelbasierte Vorwärtsbewegungsschätzung, Foreman, 30 fps, WZ-Frame 60

4.7.4. Hypothesenbasierte bidirektionale Bewegungsschätzung

Im vorangegangenen Abschnitt wurde die Dichte des Bewegungsvektorfeldes \mathbf{MVF}_f der Vorwärtsbewegungsschätzung verringert, um so eine wesentliche Reduktion der Komplexität zu erreichen. Ziel der bidirektionalen Bewegungsschätzung ist es nun, basierend auf dem bekannten Bewegungsvektorfeld \mathbf{MVF}_f , ein dichtes Bewegungsvektorfeld \mathbf{MVF} zu bestimmen, um die hohe Leistungsfähigkeit von basicPBTI zu erhalten.

Die Bestimmung eines dichten Bewegungsvektorfeldes mittels bidirektionaler Bewegungsschätzung für jedes Pixel ist aufgrund des hohen Rechenaufwandes nicht möglich. Aus diesem Grund wird hier ein zweistufiges Verfahren verwendet.

bidirektionale Bewegungsschätzung — Basierend auf einem 4×4 Raster wird die bidirektionale Bewegungsschätzung für jeden 16. Pixel durchgeführt. So erhält man die Bewegungsvektoren $\mathbf{mv}_1 \dots \mathbf{mv}_4$ des Bewegungsvektorfeldes \mathbf{MVF}_{bi} (Abb. 4.42, blau).

Hypothesenbasierte bidirektionale Bewegungsprüfung — Die weiteren Bewegungsvektoren werden nicht durch eine aufwendige Bewegungsschätzung, sondern unter Verwendung der bekannten Bewegungsvektoren aus der Nachbarschaft

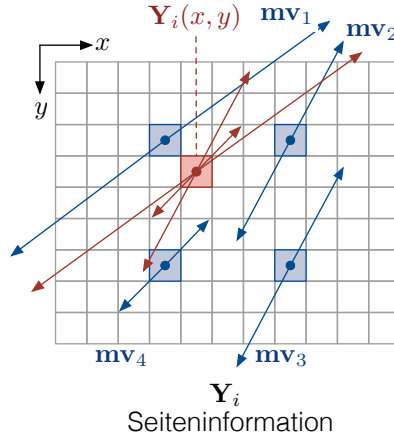


Abbildung 4.42.: Verfeinerung des Bewegungsvektorfeldes durch pixelbasierte bidirektionale Bewegungsschätzung (pBiME) für Pixel auf dem 4×4 Raster **blau**, anschließend Auswahl eines Bewegungsvektors aus den 4 bekannten Nachbarn für jeden verbleibenden Pixel **rot**

(NH) bestimmt. Es findet ausschließlich eine Prüfung statt, welcher der möglichen Vektoren $\mathbf{mv}_1 \dots \mathbf{mv}_4$ die geringsten Kosten verursacht (Gl. 4.36 und Abb. 4.42, rot).

$$\mathbf{mv} = \arg \min_{\mathbf{mv}_u} \left| \mathbf{K}_{i-1}(x - mv_{u,x}, y - mv_{u,y}) - \mathbf{K}_{i+1}(x + mv_{u,x}, y + mv_{u,y}) \right| \quad (4.36)$$

$$\mathbf{mv}_u \in \mathbf{MVF}_{bi}(x_u, y_u), \quad \mathbf{mv}_u \in NH \quad (4.37)$$

Da die Komplexität der im zweiten Schritt durchgeführten Bewegungsüberprüfung nahezu vernachlässigbar ist, sinkt die Komplexität für die bidirektionale Bewegungsschätzung um den Faktor $\frac{1}{16}$.

4.7.5. Schnelle pixelbasierte temporale Interpolation (FPBTI)

Die bisher angestellten Überlegungen zur schnellen Umsetzung der pixelbasierten temporalen Interpolation sind in dem Verfahren FPBTI (Abb. 4.43) zusammengefasst. Dies umfasst die Auswahl markanter Punkte anhand des Harris-Detektors,

gefolgt von der pixelbasierten Vorwärtsbewegungsschätzung (pFME), wobei das Fenster zur Wichtung verkleinert und die Koeffizienten so gewählt wurden, dass die notwendigen Multiplikationen durch Bitverschiebungen ersetzt werden konnten. Die bidirektionale Bewegungsschätzung wurde in zwei Stufen zerlegt, wobei erst der Bewegungsvektor für jeden 16. Pixel unter Verwendung von zwei Hypothesen bestimmt wird (pBiME). Die Schätzung der Bewegung der verbleibenden Positionen erfolgt durch Überprüfung der Bewegungen der bekannten Nachbarpunkte (pBiME2).

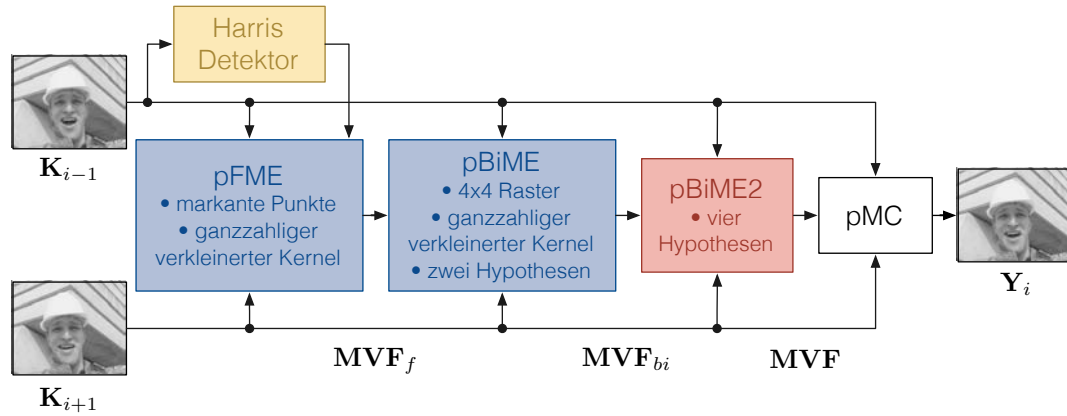


Abbildung 4.43.: Blockschaltbild der schnellen pixelbasierten zeitlichen Interpolation (FPBTI)

4.7.6. Vergleich der schnellen pixelbasierten zeitlichen Interpolation (FPBTI) mit basicPBTI

Das in diesem Abschnitt entwickelte Verfahren FPBTI soll nun anhand der Leistungsfähigkeit und Komplexität mit basicPBTI verglichen werden.

Die Abbildung 4.44 zeigt die Qualität der Seiteninformation im Vergleich zu basicPBTI für verschiedene Codierungsparameter der Key-Frames. Das entwickelte Verfahren FPBTI zeigt nur geringe Qualitätsabweichungen im Vergleich zu basicPBTI. Das gleiche Verhalten zeigt sich auch für andere Testsequenzen. Die vorgeschlagenen Ansätze zur Reduktion der Komplexität führen somit nicht zu einer signifikanten Beeinflussung der Qualität der Seiteninformation. Dies kann im Wesentlichen auf den Erhalt des dichten Bewegungsvektorfeldes zurückgeführt werden.

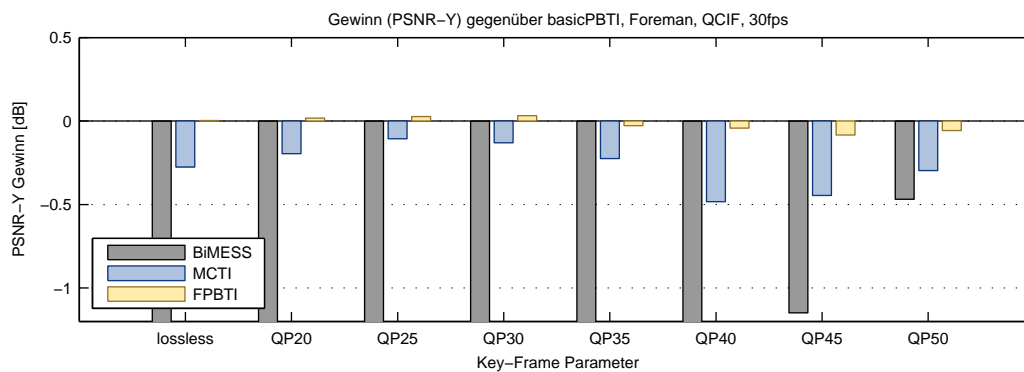


Abbildung 4.44.: Vergleich der Qualität (PSNR-Y) der Seiteninformation gegenüber basicPBTI, für verschiedene Quantisierungsparameter der Key-Frames, Foreman, QCIF, 30 fps

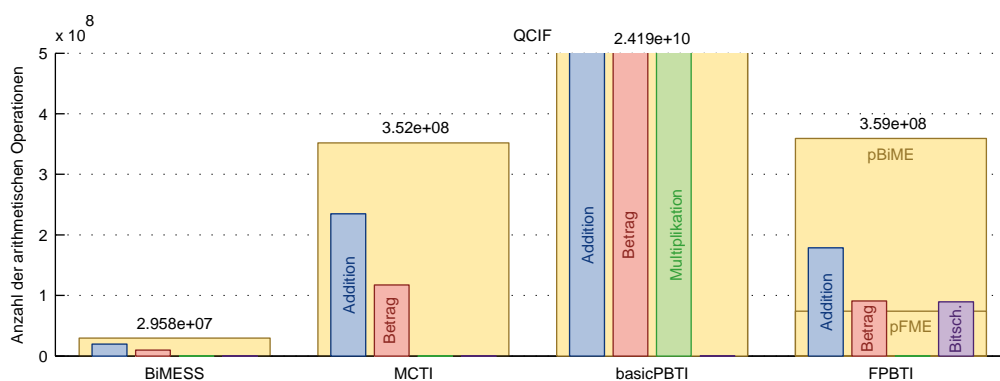


Abbildung 4.45.: Anzahl der arithmetischen Operationen im Verfahren FPBTI

Die Komplexität von FPBTI im Vergleich zu basicPBTI konnte auf ungefähr 1.5% (um $\frac{1}{66}$) reduziert werden und liegt somit auf einem Niveau anderer aus der Literatur bekannter Verfahren (BiMESS, MCTI, Abb. 4.45). Weiterhin werden in FPBTI nur sehr wenige Multiplikationen für die Umsetzung des Harris Detektors benötigt. Dieser hat eine so geringe Komplexität, dass sie nicht ins Gewicht fällt.

Eine weitere Reduktion der Komplexität könnte erreicht werden, indem die bidirektionale Bewegungsschätzung weiter optimiert wird, da diese nun die Komplexität dominiert, was vor allem durch das dichte Raster und die Halb-Pixel genaue Bewegungsschätzung zu begründen ist.

4.7.7. Fazit

Durch die schnelle Umsetzung der pixelbasierten zeitlichen Interpolation (FPBTI) konnte gezeigt werden, dass die Anzahl der notwendigen arithmetischen Operationen auf 1.5% im Vergleich zu basicPBTI reduziert werden kann, ohne dabei einen wahrzunehmenden Qualitätsverlust tragen zu müssen. Weiterhin konnten komplexe Operationen, wie Multiplikationen, weitestgehend durch einfache Bitverschiebungen ersetzt werden.

Die Herausforderung für FPBTI besteht in der Interpolation von Sequenzen mit sehr schneller oder stark inhomogener Bewegung. Diese Probleme wurden in den Abschnitten 4.4 und 4.5 gelöst. Die hier vorgeschlagenen Ansätze zur Komplexitätsreduktion können auch auf GpTIE und MPBTI sowie auf APBTI(2) übertragen werden. Die genannten Verfahren wurden entworfen, um eine gute Seiteninformationsqualität für Sequenzen mit schneller und/oder inhomogener Bewegung zu erzielen.

4.8. Zusammenfassung

Da die Qualität der Seiteninformation starken Einfluss auf die Leistungsfähigkeit des Videocodierungssystems hat, wurden in diesem Kapitel Probleme in Bezug auf die Generierung der Seiteninformation analysiert und geeignete Verfahren zur Effizienzsteigerung entwickelt.

Im Fokus stand die temporale Interpolation unter Berücksichtigung von schnellen und inhomogenen Bewegungen. Diese Probleme konnten durch einen zweistufigen Ansatz aus robuster Schätzung der globalen Bewegung sowie geführter pixelbasierter Bewegungsschätzung und multiplen Bewegungshypothesen gelöst werden. Neben der temporalen Interpolation wurde eine adaptive temporale Extrapolation in Bildrandbereichen eingeführt, welche durch die Interpolation nicht abgedeckt werden können. Darüber hinaus wurde auch die Problematik von Auf- sowie Verdeckungen auf der gesamten Bildfläche behandelt und durch eine adaptive zeitliche Extrapolation und durch eine Extrapolation des Bewegungsvektorfeldes gelöst.

Die Mehrzahl der entwickelten Verfahren basieren auf der pixelbasierten Bewegungsschätzung und -kompensation, wodurch die Ausbildung von Blockartefakten vermieden wird. Diese Verfahren haben aufgrund des dichten Bewegungsvektorfeldes jedoch eine sehr hohe Komplexität, welche, durch die Auswahl markanter Bildbereiche sowie geschickter Interpolation des Bewegungsvektorfeldes, signifikant reduziert werden konnte.

Die Grundidee hinter den entwickelten Verfahren besteht darin, markante Bereiche im Bild geschickt zu nutzen, um die Leistungsfähigkeit der Bewegungsschätzung zu verbessern und um die Komplexität zu verringern.

5

Kapitel 5

Wyner-Ziv Codierung – *oDeViCe*

Der zweite Teil des entwickelten DVC-Systems wird in diesem Kapitel beschrieben. Dieser bezieht sich auf die Quellencodierung, was sowohl die verlustlose Slepian-Wolf Codierung als auch die darauf aufbauende verlustbehaftete Wyner-Ziv Codierung mit einschließt.

Neben der Generierung der Seiteninformation (Kap. 4) ist die Wyner-Ziv Codierung die zweite wichtige Komponente in einem System zur Verteilten Videocodierung. Wyner-Ziv Codierung beschreibt die verlustbehaftete Quellencodierung. Diese wird, in Anlehnung an konventionelle Codierungsverfahren, durch eine Quantisierung und nachfolgende verlustlose Quellencodierung umgesetzt (Abb. 3.2, S. 39). In diesem Fall wird zur verlustlosen Codierung die Slepian-Wolf Codierung eingesetzt. Diese zählt zur Gruppe der Entropiecodierung und stellt somit das Äquivalent zur Lauflängencodierung, Huffman-Codierung oder zur arithmetischen Codierung in konventionellen Videocodierungssystemen dar.

Im Folgenden soll die Slepian-Wolf Codierung im Detail betrachtet und auf Problemstellungen wie die Terminierung, sichere Decodierung und ungleichmäßig verteilte Eingangssymbole eingegangen werden. Darauf aufbauend wird die verwendete Wyner-Ziv Codierung vorgestellt. Besondere Beachtung findet hier die Binarisierung, die Quantisierung sowie die Modellierung der statistischen Abhängigkeiten in einem Verteilten Videocodierungssystem. Abschließend wird der Einfluss des für

die Codierung der Key-Frames eingesetzten Codecs auf das Gesamtsystem analysiert.

5.1. Slepian-Wolf Codierung

Für die Verteilte Videocodierung spielt die asymmetrische Slepian-Wolf Codierung (Abb. 2.9, S. 20), bei der die Seiteninformation Y am Decoder vorliegt, eine wichtige Rolle. Technisch umsetzen lässt sich die Slepian-Wolf Codierung zum Beispiel durch die Verwendung leistungsfähiger Kanalcodierungsverfahren (Turbocode, LDPC Code).

5.1.1. Turbocodebasierte Slepian-Wolf Codierung

Die Slepian-Wolf Codierung kann, wie in der Literatur beschrieben [Aaron u. a. 2002][Varodayan u. a. 2005], mittels leistungsfähiger Kanalcodierungsverfahren umgesetzt werden. Im Wesentlichen werden dazu in der Literatur leistungsfähige Turbo- oder LDPC-Codierungsverfahren eingesetzt. In diesem Abschnitt wird die Slepian-Wolf Codierung auf Basis von Turbocodes betrachtet. Weiterführend wird ein Vergleich zur SW-Codierung basierend auf LDPCA Codes gezogen (Abschn. 5.1.2).

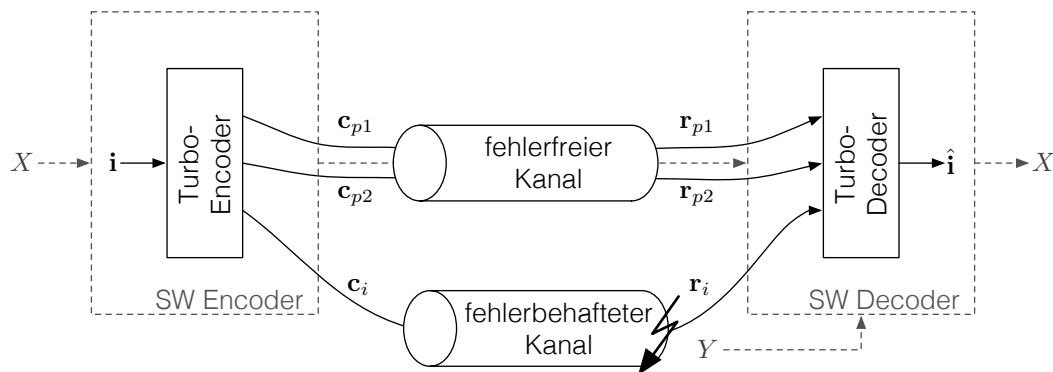


Abbildung 5.1.: Umsetzung der Slepian-Wolf Codierung mittels eines systematischen Kanalcodes (z.B. Turbo-Codierung)

Die Abbildung des Slepian-Wolf Codierungsproblems auf einen systematischen Kanalcode (wie z.B. Turbocode, LDPC Code) ist in Abbildung 5.1 dargestellt. Dabei wird der systematische Anteil \mathbf{c}_i des Codevektors \mathbf{c} nicht direkt übertragen, sondern

über die bekannte Seiteninformation Y am Empfänger gewonnen. Die statistische Abhängigkeit zwischen den beiden Quellen X und Y wird durch einen fehlerbehafteten Kanal modelliert. Die punktierten Paritätsbitvektoren \mathbf{c}_{p1} , \mathbf{c}_{p2} werden fehlerfrei übertragen und dienen der Korrektur der Seiteninformation, sodass die Originalnachricht rekonstruiert werden kann.

Die Menge an übertragenen Paritätssymbolen \mathbf{c}_{p1} , \mathbf{c}_{p2} pro Informationssymbol gibt die notwendige Datenrate R zur erfolgreichen Decodierung an. Diese hängt von der Korrelation zwischen X und Y ab. In der Kanalcodierung wird anstatt der Datenrate R (auch Quellenrate $R_s = R = \frac{n-k}{k}$) die Coderate R_c verwendet (Gl. 2.5, S. 11). Die Coderate ist jedoch im Rahmen der Verteilten Quellencodierung nicht intuitiv und wird aus diesem Grund hier nicht weiter verwendet.

Der im Rahmen dieser Arbeit verwendete parallel verkettete Faltungscodier ist [Rowitch u. Milstein 2000] (RCPT - *rate compatible punctured turbo codes*) entnommen. Dessen Datenrate R kann entsprechend der Korrelation (bedingten Entropie $H(X|Y)$) durch Punktierung angepasst werden. Als Generator der RSC Codes wird [1, 33/31] verwendet (z-Transformierte des Generators, siehe Gl. 5.1, 5.2, $D = z^{-1}$)

$$g_1(D) = 1 \tag{5.1}$$

$$g_2(D) = \frac{1 + D + D^3 + D^4}{1 + D + D^4} \tag{5.2}$$

Im Allgemeinen werden Faltungscodes bei der Fehlerschutzcodierung terminiert. In [Rowitch u. Milstein 2000] wurde angemerkt, dass die Terminierung der RCPT Codes zu keinem Vorteil bei der Fehlerschutzcodierung führt. Der Einfluss verschiedener Terminierungsmethoden auf die Leistungsfähigkeit von RCPT Codes für die Slepian-Wolf Codierung soll im Folgenden weiterführend betrachtet werden.

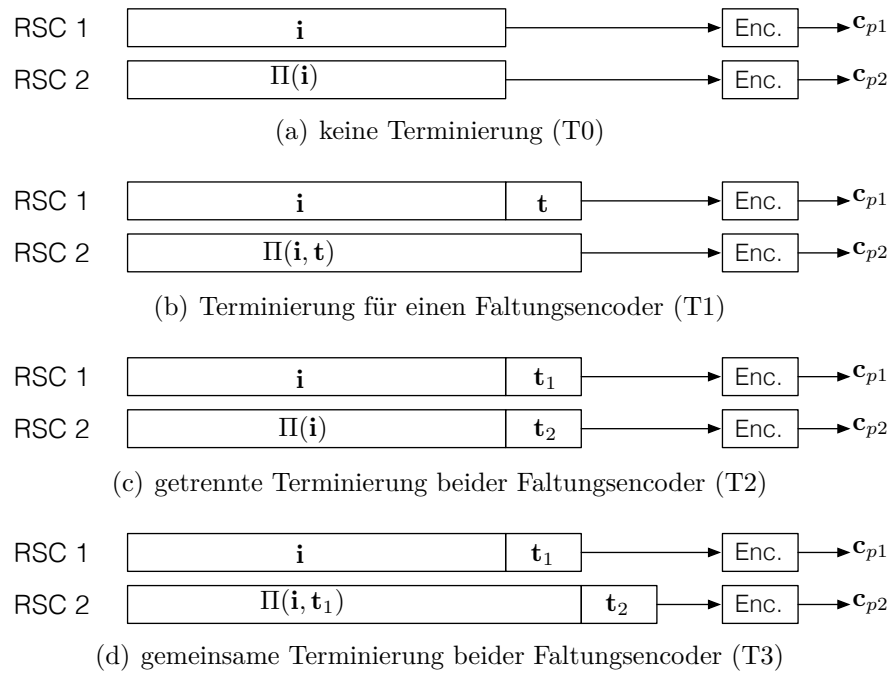


Abbildung 5.2.: Methoden zur Terminierung eines Turbocodes

Einfluss der Terminierung des Turbocodes auf die Slepian-Wolf Codierperformanz

Faltungscodes werden in der Regel terminiert, um deren Leistungsfähigkeit zu erhöhen. Unter Terminierung versteht man dabei das Anfügen von sogenannten Terminierungssymbolen (*tailbits*, t), wodurch der Encoder zum Ende der Codierung einen festen Zustand einnimmt. Der hier verwendete Turbocode besteht aus zwei parallel verketteten rekursiven Faltungscodes (RSC). Die Frage der richtigen Terminierung, um eine hohe Leistungsfähigkeit für die Slepian-Wolf Codierung zu erhalten, wird in diesem Abschnitt untersucht.

Dazu werden die in Abbildung 5.2 dargestellten Terminierungsmethoden betrachtet. Die zusätzlich übertragenen Terminierungssymbole (*tailbits*) werden nicht punktiert.

keine Terminierung (T0) — Es werden weder für den ersten noch den zweiten Faltungscoder (RSC) Terminierungssymbole angefügt.

Terminierung eines Faltungscoders (T1) — Der erste Faltungscoder (RSC 1) wird terminiert, wobei die Terminierungssymbole t im zweiten Faltungscoder (RSC

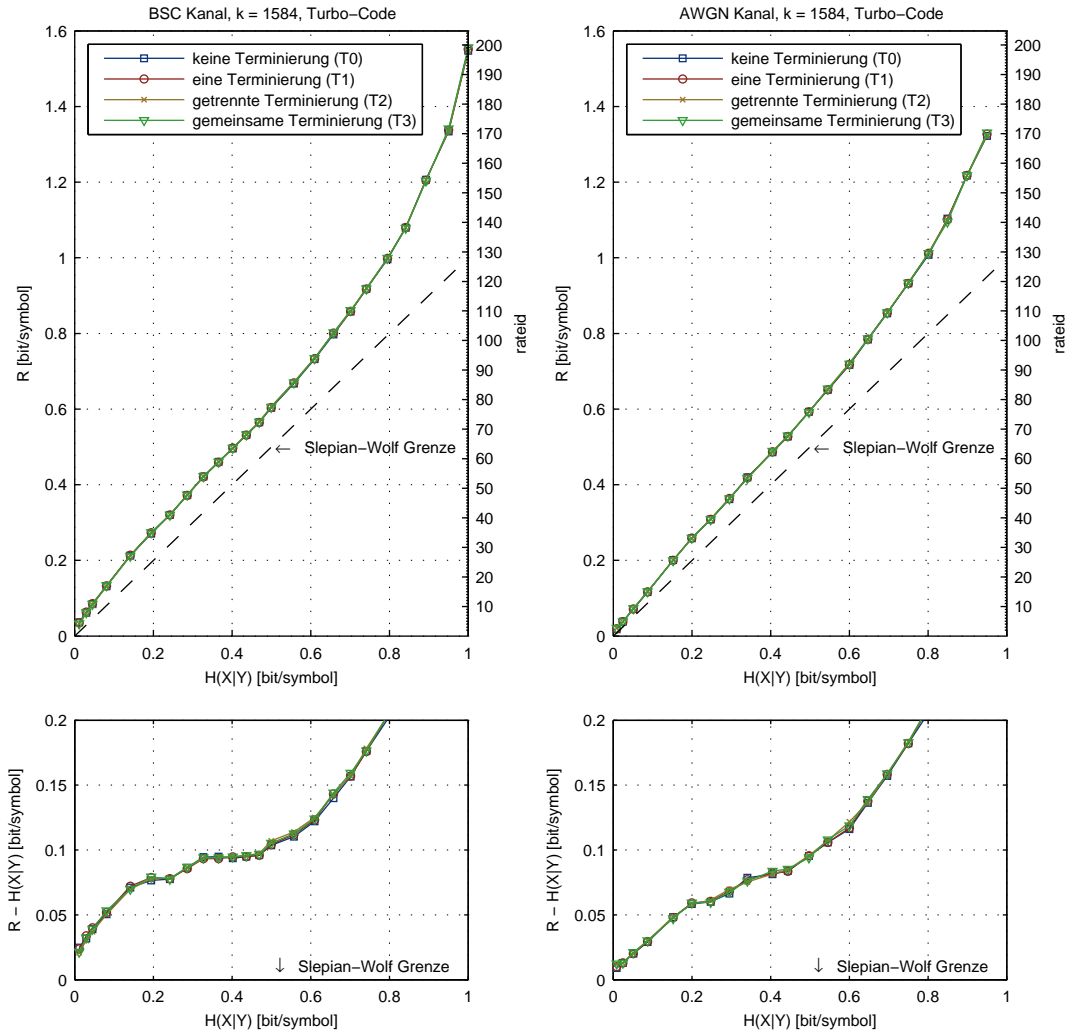


Abbildung 5.3.: Performanz der Slepian-Wolf Codierung auf Basis eines Turbocodes unter Verwendung verschiedener Terminierungsarten, BSC und AWGN Kanal, (oben) absolute Datenrate, (unten) Differenz zur Slepian-Wolf Grenze

2) mit codiert werden. Der RSC 2 wird nicht terminiert.

getrennte Terminierung beider Faltungscodes (T2) — Für beide Faltungscodes werden die Terminierungssymbole getrennt voneinander bestimmt.

gemeinsame Terminierung beider Faltungscodes (T3) — Wie T1, jedoch wird auch der zweite Faltungscode (RSC 2) anschließend terminiert.

Abbildung 5.3 zeigt die notwendige Datenrate zur erfolgreichen Slepian-Wolf Deco-

dierung im Vergleich zur bedingten Entropie $H(X|Y) = H(X) - I(X, Y)$ (Gl. 2.9, 2.10). Die Korrelation zwischen den Symbolen der Quelle X und Y wurden durch einen BSC- bzw. AWGN-Kanal modelliert, wobei binäre gleichverteilte Eingangssymbole verwendet wurden. Die Simulationsergebnisse wurden über 1000 Iterationen gemittelt, wobei $k = 1584$ Informationssymbole je Iteration verarbeitet wurden.

Gut zu erkennen ist, dass die Art der Terminierung keinen signifikanten Einfluss auf die Leistungsfähigkeit der Slepian-Wolf Codierung mittels Turbocodes hat. Aus diesem Grund wird im Weiteren ausschließlich die Methode ohne Terminierung (T0) verwendet.

In [Rowitch u. Milstein 2000] wurde herausgestellt, dass die Terminierung auch zu einer Verschlechterung der Leistungsfähigkeit von RCPT Codes führen kann (Fehlerschutzcodierung). Diese Aussage kann für die Slepian-Wolf Codierung nicht übernommen werden, da sich gezeigt hat, dass die Terminierung im Kontext der SW-Codierung keinen Einfluss auf die Leistungsfähigkeit hat (Abb. 5.3).

Bestimmung des Punktierungsschemas

In [Varodayan u. a. 2005] wurden LDPCA Codes (*low density parity check accumulate code*) für die SW-Codierung entwickelt. Diese wurden so entworfen, dass bei einer Datenrate von $R = 1$ bit/symbol (Coderate $R_c = \frac{1}{2}$) eine erfolgreiche Decodierung sichergestellt ist. Dies wird erreicht indem die Teilmatrix \mathbf{A} der Prüfmatrix \mathbf{H} (Abschn. 2.1.2, S. 11) vollen Rang besitzt (\mathbf{A} ist reguläre Matrix). Hiermit kann die Dekodierung durch lösen des Gleichungssystems $\mathbf{c}_p = \mathbf{i} \cdot \mathbf{A}$ (Gl. 2.7, S. 11) unabhängig von der Seiteninformation Y erreicht werden, sofern mindestens k Paritätssymbole bei k Informationssymbolen übertragen wurden.

Dieser Ansatz soll nun auf Turbocodes übertragen werden. Dazu muss das Punktierungsschema \mathbf{P} und der Interleaver Π entsprechend gewählt werden. Die Idee besteht darin, das Punktierungsschema schrittweise zu erzeugen, wobei \mathbf{A} in jedem Schritt so erweitert wird, dass diese vollen Rang besitzt. Diese Herangehensweise ist in folgendem Algorithmus beschrieben (Abb. 5.4):

1. Erzeugung der $2k \times k$ Matrix \mathbf{A}_{full} auf Basis des Generators $g_2(D)$ und des Interleavers Π für den unpunktierten Fall.

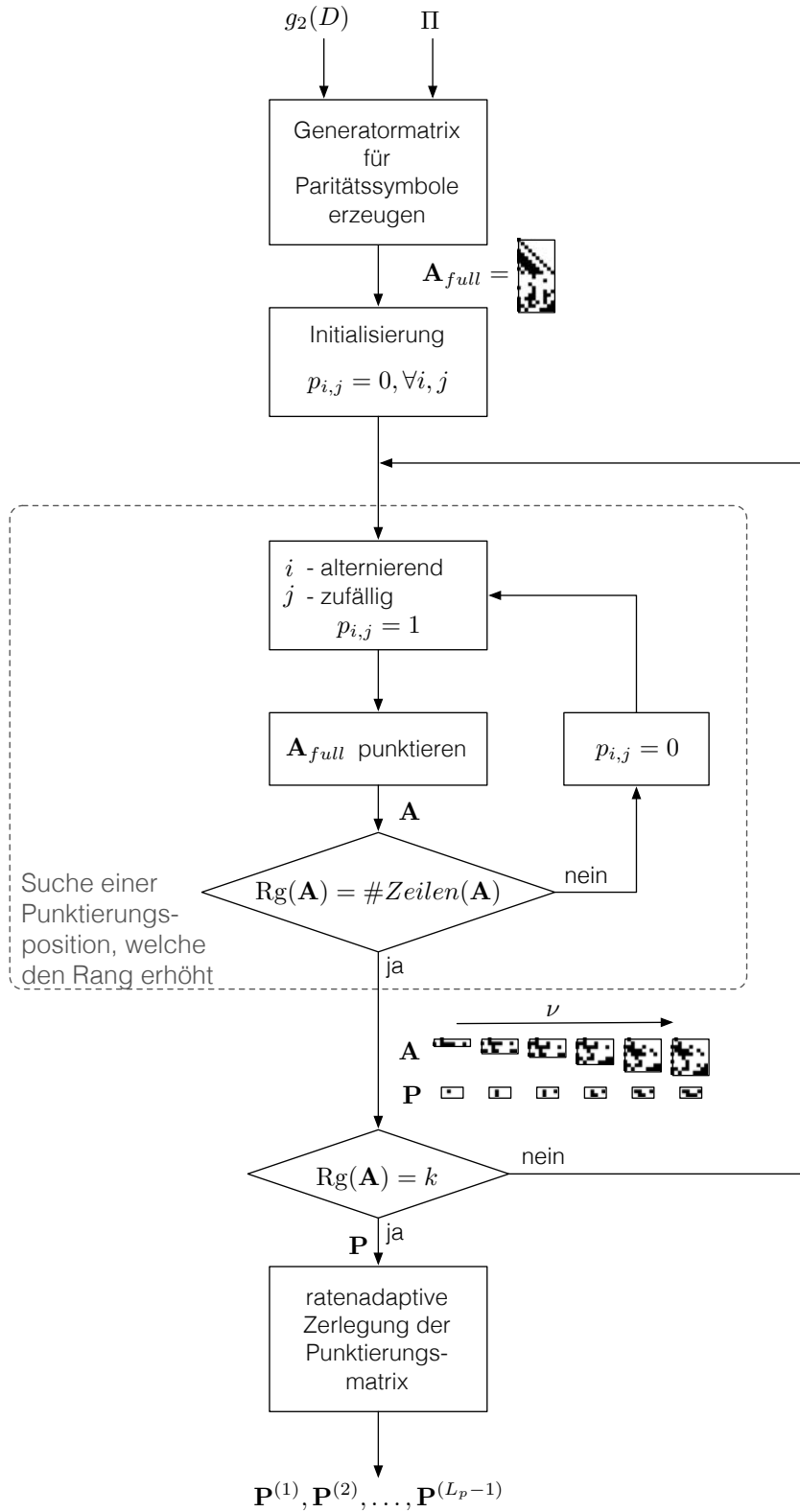


Abbildung 5.4.: Generierung des ratenadaptiven Punktierungsschemas mit vollem Rang der Matrix \mathbf{A} bei der Rate $R = 1$ bit/symbol, $R_c = \frac{1}{2}$, Parameter der angegebenen Beispiele: $k = 10$, $L_p = 6$

2. Alle Paritätssymbole punktieren $p_{i,j} = 0$.
3. Zufällig eine Position in der Punktierungsmatrix \mathbf{P} wählen, welche nicht punktiert wird, wobei der systematische Anteil immer punktiert wird. Weiterhin wird im ersten Schritt jeweils eine nicht-punktierte Position für die Codevektoren \mathbf{c}_{p1} und \mathbf{c}_{p2} gewählt, um die Turbo-Decodierung bei der geringsten Datenrate R zu ermöglichen. In den weiteren Schritten werden alternierend $i \in \{2; 3\}$ unpunktierte Positionen gewählt.
4. Erzeugen der Matrix \mathbf{A} für das aktuelle Punktierungsschema durch Auswahl der nicht punktierten Zeilen der Matrix \mathbf{A}_{full} .
5. Prüfen, ob die Matrix \mathbf{A} höchstmöglichen Rang besitzt, wenn nicht wird die als Letztes gewählte Position in der Punktierungsmatrix wieder gelöscht und eine andere gewählt (weiter mit 3.).
6. Prüfen, ob \mathbf{A} einen vollen Rang hat, wenn nicht, dann weiter mit 3.
7. Zerlegung der Punktierungsmatrix \mathbf{P} in ein ratenadaptives Punktierungsschema $\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \dots, \mathbf{P}^{(L_p-1)}$, indem zufällig je Ratenstufe weitere nicht-punktierte Positionen aus \mathbf{P} gewählt werden.

Die Elemente der Matrix \mathbf{A}_{full} und somit auch der punktierten Matrix \mathbf{A} können auch direkt durch Anwendung von Gleichung 5.3 bestimmt werden, wobei \mathbf{c} den Codevektor des ersten Faltungscoders (nicht interleaved) wiedergibt, wenn der Informationsvektor $\mathbf{i} = (1, 0, 0, \dots, 0)$ codiert wird.

$$a_{i,j} = \begin{cases} c_{i-j+1} & : i \leq k \\ c_{i-k-\pi^{-1}(j)+1} & : i > k \end{cases} \quad (5.3)$$

wobei π^{-1} den Index zum Deinterleaving angibt ($b_{\pi^{-1}(j)} = b'_j$, $\mathbf{b} = \Pi^{-1}(\mathbf{b}')$).

Der hier vorgestellte Algorithmus stellt sicher, dass ein Punktierungsschema gewählt wird, welches bei der Datenrate $R = 1$ bit/symbol zu einer Matrix \mathbf{A} mit vollem Rang führt, wodurch die erfolgreiche Decodierung sichergestellt ist. Aufgrund der Periodizität der Punktierung (bzw. kleine Punktierungsperiode L_p) kann die

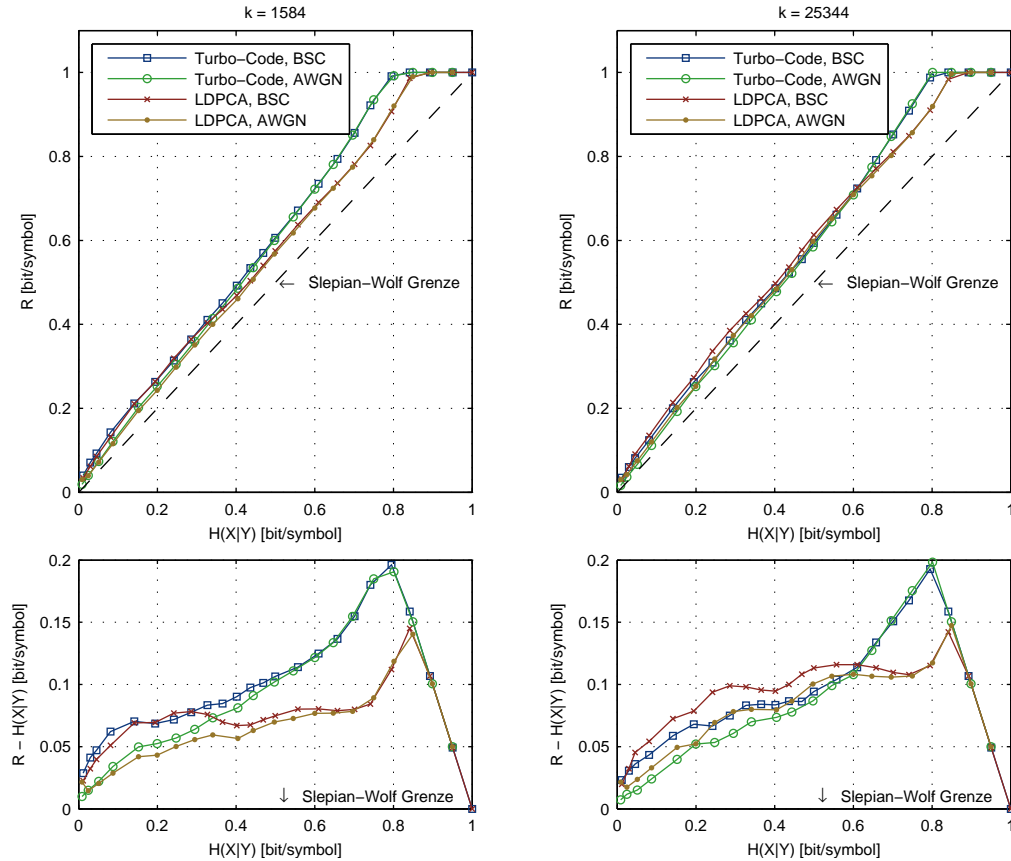


Abbildung 5.5.: Vergleich von turbocodebasierter und LDPCA basierter Slepian-Wolf Codierung für verschieden lange Symbolfolgen, irregulärer LDPCA Code mit Knotengraden 2 bis 8

Datenrate nicht beliebig fein eingestellt werden, wodurch die Datenrate zur sicheren Decodierung im praktischen Einsatz etwas größer als 1 bit/symbol ist und somit die Matrix \mathbf{A} mehr als k Zeilen besitzt.

Es sei angemerkt, dass die Geschwindigkeit des Algorithmus gesteigert werden kann, indem die Berechnung von $\text{Rg}(\mathbf{A})$ nicht in jeder Iteration neu durchgeführt wird, sondern die entsprechenden Zwischenergebnisse, wie z.B. die Trapezmatrix, der letzten erfolgreichen Iteration weiter verwendet werden.

In Abbildung 5.5 kann die Leistungsfähigkeit der so erzeugten Kombination aus Interleaver Π und Punktierungsschema \mathbf{P} abgelesen werden. Die Datenrate von 1 bit/symbol wird hierbei nur geringfügig überschritten, was mit der beschränkten Punktierungsperiode L_p zu begründen ist.

5.1.2. Vergleich von LDPCA basierter und turbocodebasierter Slepian-Wolf Codierung

Im Abschnitt 5.1.1 wurde die Slepian-Wolf Codierung, basierend auf ratenadaptiven Turbocodes (RCPT), beschrieben. In der Literatur werden auch entsprechend angepasste LDPCA Codes zur Slepian-Wolf Codierung eingesetzt [Varodayan u. a. 2005].

Der Vergleich zwischen den RCPT und LDPCA Codes soll als Grundlage dienen, um eine Entscheidung für die Umsetzung der SW-Codierung zur Verteilten Videocodierung zu treffen.

Die Codierungsperformanz der Turbo- und LDPCA Codes auf dem BSC sowie AWGN Kanal ist in Abbildung 5.5 dargestellt. Für eine geringe Korrelation sieht man deutlich ($H(X|Y) > 0.4$ bit/symbol), dass die Datenrate unter Verwendung eines LDPCA Codes für eine kleine Anzahl von Symbolen $k = 1584$ geringer ist. Hingegen unterscheidet sich die Leistungsfähigkeit der Turbocodes gegenüber LDPCA Codes, in Bereich starker Korrelation ($H(X|Y) < 0.4$ bit/symbol) nur gering (2% geringere bis 4% höhere Datenrate). Für eine hohe Anzahl von Symbolen ($k = 25344$), wie sie bei der pixelbasierten DVC auftritt, erreicht die turbocodebasierte Slepian-Wolf Codierung eine höhere Leistungsfähigkeit, im Bereich der starken Korrelation im Vergleich zu dem LDPCA Code.

Für die Verteilte Videocodierung ist insbesondere der Bereich starker Korrelation zwischen Originalsignal X und der Seiteninformation Y von Interesse. In diesem Bereich gibt es nur geringe Performanzunterschiede ($k = 1584$) zwischen Turbocode und LDPCA Codes. Für lange Symbolfolgen ($k = 25344$) zeigen die Turbocodes sogar eine bessere Leistungsfähigkeit als LDPCA Codes. Basierend auf diesen Ergebnissen wird für die weitere Arbeit ausschließlich auf Turbocodes zurückgegriffen.

5.1.3. Ungleichmäßig verteilte Quellensymbole

Bisher wurde die Leistungsfähigkeit der turbocodebasierten Slepian-Wolf Codierung für binäre Quellen mit gleichverteilten Symbolen betrachtet. Weiterführend

soll untersucht werden, inwieweit sich nicht gleichverteilte Eingangssymbole x_i auf die Performanz der Slepian-Wolf Codierung auswirken.

Abbildung 5.6(a) zeigt die SW-Codierungsperformanz für Turbocodes bei gleichverteilten $H(X) = 1$ bit/symbol und ungleichmäßig verteilten Symbolen ($H(X) = 0.47$ bit/symbol, $\Pr(X = 0) = 0.1$). Der Decoder verfügt in einem Fall über Kenntnis der ungleichmäßigen Verteilung und somit über a-priori Information, welche durch das Log-Likelihood-Verhältnis L_a (Gl. 5.4) abgebildet wird.

$$L_a(\hat{x}_i) = \ln \frac{\Pr(x_i = 0)}{\Pr(x_i = 1)} \quad (5.4)$$

Deutlich zu erkennen ist, dass bei fehlender Kenntnis über die Ungleichverteilung eine starke Reduktion der Leistungsfähigkeit einsetzt. Es besteht somit eine große Lücke zwischen der Slepian-Wolf Grenze und der vom Codec erreichten Performanz. Verfügt der Decoder über die entsprechende a-priori Information, so kann eine Leistungsfähigkeit erreicht werden, die teilweise besser ist als im Fall gleichverteilter Symbole.

In der konventionellen Quellencodierung wird zur Kompression ungleich verteilter Symbole zum Beispiel die Huffman- oder arithmetische Codierung eingesetzt. Diese können jedoch nicht die Korrelation zu der Seiteninformation ausnutzen. In Abbildung 5.6(a) ist exemplarisch die entsprechende Datenrate bei Verwendung einer arithmetischen Codierung dargestellt. Diese erreicht für geringe Korrelationen eine bessere Leistungsfähigkeit als die SW-Codierung, wohingegen bei starker Korrelation die SW-Codierung eine wesentlich bessere Leistungsfähigkeit zeigt.

Übertragung der Quellenstatistik, AIT

Wie zuvor gezeigt wurde, muss der Slepian-Wolf Decoder Kenntnis über ungleichmäßig verteilte Quellensymbole haben, um eine hohe Leistungsfähigkeit zu erreichen. Eine Möglichkeit, dies zu erreichen, besteht in der Bestimmung des a-priori Log-Likelihood Verhältnisses L_a^{AIT} am Encoder, deren Quantisierung (`quand(...)`) und der Übertragung zum Decoder (AIT, *a-priori information transfer*, Abb. 5.7(a)).

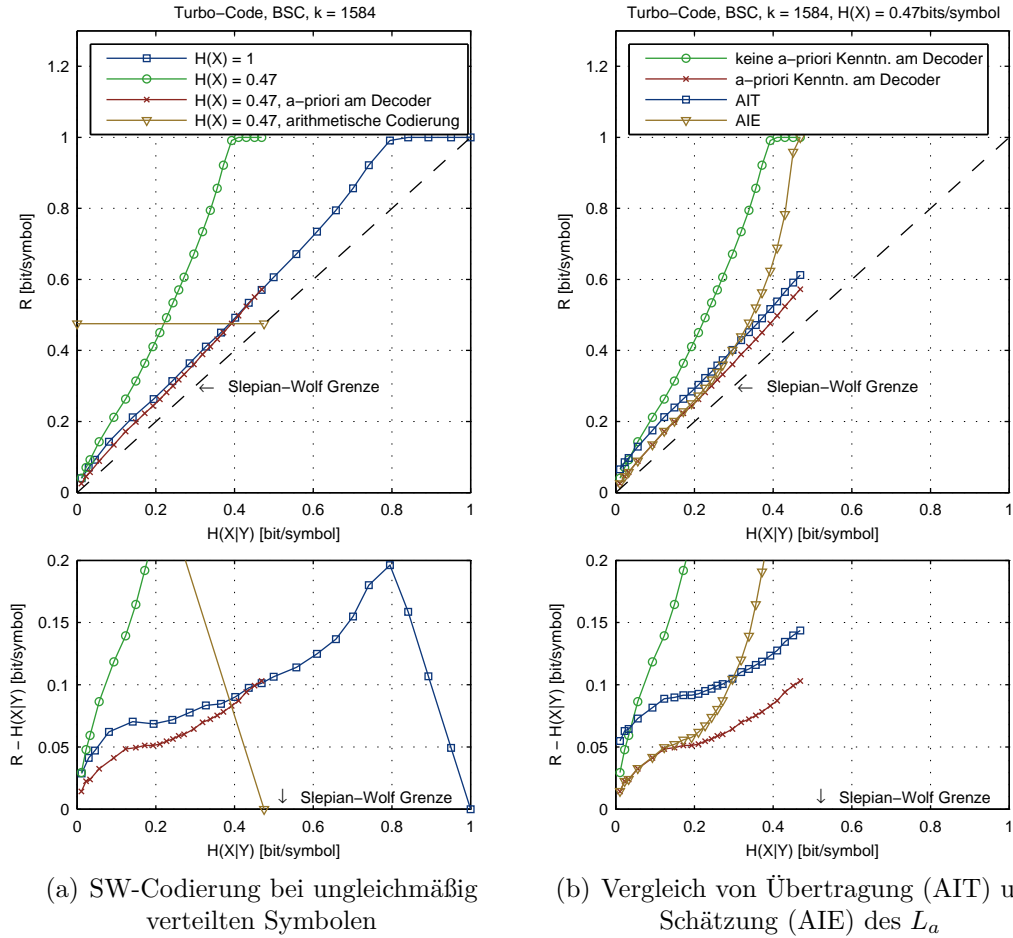


Abbildung 5.6.: Performanz der SW-Codierung (Turbocode) bei ungleichmäßig verteilten Eingangssymbolen, (a) ohne und mit (a-priori) Kenntnis am Decoder, (b) sowie der Übertragung (AIT) oder Schätzung (AIE) des a-priori Log-Likelihood-Verhältnisses L_a

$$L_a^{AIT}(\hat{x}_i) = \text{quand} \left(\ln \frac{\Pr(x_i = 0)}{\Pr(x_i = 1)} \right) \quad (5.5)$$

Somit verfügt der Decoder über Kenntnis der ungleichmäßigen Verteilung der Eingangssymbole. Wie die Ergebnisse in Abbildung 5.6(b) zeigen, wird die Leistungsfähigkeit verringert, da zur Übertragung von L_a^{AIT} eine zusätzliche Datenrate R^{AIT} notwendig ist. Es wurde hier die Repräsentation des a-priori Log-Likelihood Verhältnisses als eine 64 bit-Gleitkommadarstellung angenommen. Bei steigender Anzahl von Symbolen k wird die Leistungsfähigkeit gesteigert, da die Rate R^{AIT} je

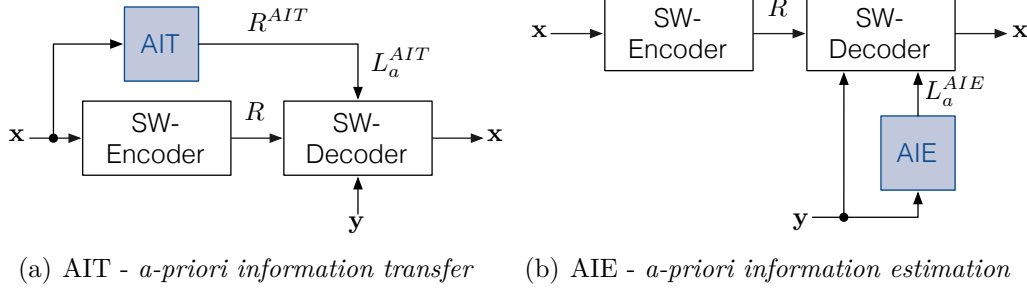


Abbildung 5.7.: Bestimmung der a-priori Information für die Slepian-Wolf Codierung ungleichmäßig verteilter Quellen

Symbol verringert wird.

Schätzung der Quellenstatistik am Decoder, AIE

Neben der Übertragung des a-priori Log-Likelihood Verhältnisses kann dies auch am Decoder bestimmt werden. Bei starker Korrelation zwischen X und Y ist die Verteilung der Seiteninformation Y ähnlich der Verteilung der Eingangssymbole X . Somit kann das a-priori Log-Likelihood Verhältnis L_a^{AIE} näherungsweise aus der Verteilung der Seiteninformation bestimmt werden (AIE, *a-priori information estimation*, Abb. 5.7(b)).

$$L_a^{AIE}(\hat{x}_i) = \ln \frac{\Pr(y_i = 0)}{\Pr(y_i = 1)} \approx \ln \frac{\Pr(x_i = 0)}{\Pr(x_i = 1)} \quad (5.6)$$

Dieses Verfahren besitzt gegenüber AIT den Vorteil, dass keine zusätzliche Übertragung notwendig ist. Die Leistungsfähigkeit ist für eine starke Korrelation zwischen X und Y sehr hoch, wohingegen bei geringer Korrelation nur die Leistungsfähigkeit, wie im Fall fehlender Kenntnis der a-priori Information am Decoder erreicht wird (Abb. 5.6(b)).

5.1.4. Fazit

In diesem Abschnitt wurde die Slepian-Wolf Codierung auf Basis der Turbo-Codierung beschrieben und unter anderem die Art der Terminierung der verketteten

Faltungscodes betrachtet. Dabei wurde festgestellt, dass kein Einfluss auf die Leistungsfähigkeit der Slepian-Wolf Codierung besteht, somit wurde die Terminierungsart T0 (keine Terminierung) für *ρDeViCe* gewählt.

Weiterhin wurde ein Verfahren vorgestellt, welches es erlaubt, den Interleaver Π und das Punktierungsschema \mathbf{P} so aufeinander abzustimmen, dass eine erfolgreiche Decodierung bei einer Datenrate von $R = 1$ bit/symbol (Coderate $R_c = \frac{1}{2}$) sichergestellt ist. Für LDPCA Codes war dieser Ansatz schon bekannt [Varodayan u. a. 2005].

Die so erzeugten ratenadaptiven Turbocodes (RCPT) zeigen eine ähnliche Leistungsfähigkeit zur Slepian-Wolf Codierung wie LDPCA Codes. Für große Blocklängen ($k = 25344$) zeigen die RCPT Codes, in den für die Verteilte Videocodierung relevanten Bereichen, eine bessere Leistungsfähigkeit als LDPCA Codes.

Abschließend wurde untersucht, wie die Leistungsfähigkeit der Slepian-Wolf Codierung bei ungleichmäßig verteilten Eingangssymbolen verbessert werden kann. Dieses Problem wird in konventionellen Systemen durch die Entropiecodierung gelöst. Hierbei wurden zwei Verfahren (AIE, AIT) vorgeschlagen, welche a-priori Information am Decoder zur Verfügung stellen. Mit beiden Verfahren konnte eine wesentliche Verbesserung der Leistungsfähigkeit erreicht werden, wobei AIE bei starker Korrelation zwischen Seiteninformation Y und Original X vorzuziehen ist.

5.2. Wyner-Ziv Codierung

Die Wyner-Ziv Codierung stellt im Gegensatz zur Slepian-Wolf Codierung die verlustbehaftete Verteilte Quellencodierung zur Verfügung. In den folgenden Abschnitten wird die Umsetzung der WZ-Codierung beschrieben. Zusätzlich wird auf die Modellierung der Abhängigkeit zwischen Seiteninformation Y und Original X eingegangen und die Anwendung des Kanalmodells in der Decodierung dargestellt.

5.2.1. Wyner-Ziv System

Das verwendete System zur Wyner-Ziv Codierung (Abb. 5.8) besteht neben der bekannten Slepian-Wolf Codierung aus der Quantisierung (Q), der Bitebenenextraktion und -rekombination (BPE/BPC) sowie der Rekonstruktion (R). Des Weiteren bildet das Kanalmodell (Abschn. 5.2.2) die Abhängigkeit zwischen den Seiteninformationssymbolen \mathbf{y} , den Quantisierungssymbolen \mathbf{q} und den Originalsymbolen \mathbf{x} ab und stellt Zuverlässigkeitsinformationen für die SW-Decodierung zur Verfügung.

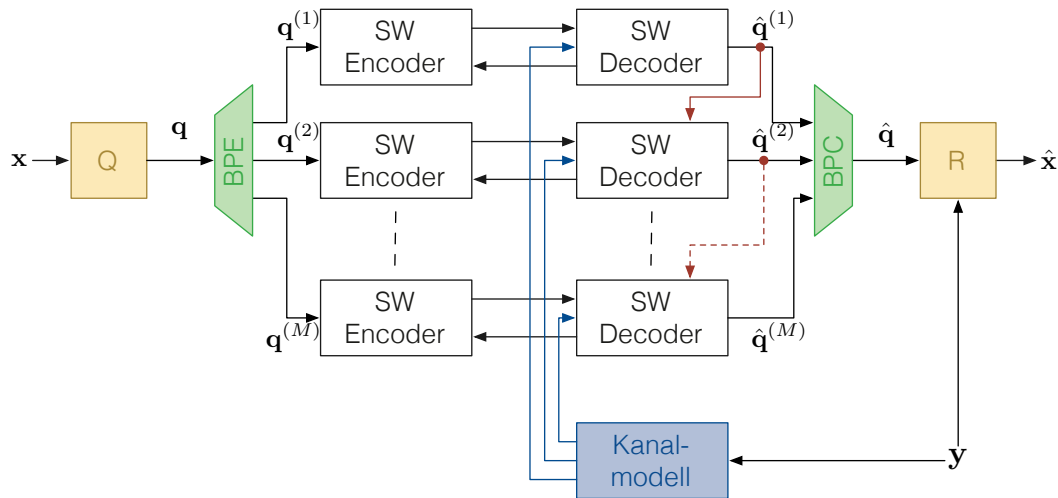


Abbildung 5.8.: Umsetzung der Wyner-Ziv Codierung durch Quantisierung (Q), Slepian-Wolf Codierung (SW) und Rekonstruktion (R)

Pixelauswahlreihenfolge

Im ersten Schritt müssen die Bildpunkte im Originalbild \mathbf{X} in den Vektor \mathbf{x} überführt werden. Dazu werden hier drei unterschiedliche Ansätze betrachtet (Abb. 5.9). Dies ist zum einen das oft angewendete zeilenweise Auslesen der Pixel. Darüber hinaus werden das blockweise Auslesen und das Auswählen in zufälliger Reihenfolge betrachtet.

Die Simulationsergebnisse haben gezeigt, dass die RD-Performanz der WZ-Frames am höchsten ist, wenn die zufällige Auswahlreihenfolge verwendet wird (Abb. 5.10). Dies ist durch die blockartige Fehlerstruktur der Seiteninformation zu erklären, welche von Faltungscodes (wie in SW-Codierung eingesetzt) nicht gut korrigiert

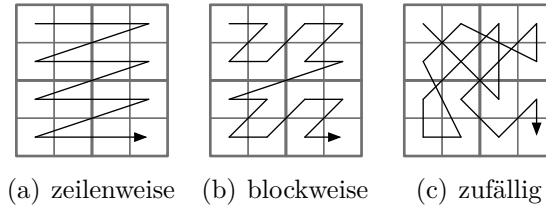


Abbildung 5.9.: Auswahlreihenfolge (*scan order*) der Pixel im Bild

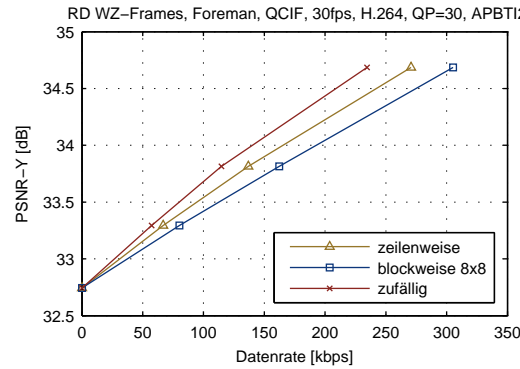


Abbildung 5.10.: Vergleich der RD -Performanz der WZ-Frames bei unterschiedlichen Pixelauswahlreihenfolgen

werden können. Durch die zufällige Auswahlreihenfolge werden die blockartigen Fehler zerstreut. Da die Auswahlreihenfolge für alle Bilder gleich ist, muss diese nicht gesondert übertragen werden.

Quantisierung

Die Grauwerte \mathbf{x} des Originalbildes werden mittels linearer Quantisierung (2^M Stufen) in die Quantisierungssymbole \mathbf{q} überführt (Gl. 5.7). Weiterführend wird in Abschnitt 5.2.4 auf die adaptive, an die Verteilung der Grauwerte angepasste, Quantisierung eingegangen, welche allgemein durch Gleichung 5.8 beschrieben werden kann.

$$q_i = \left\lfloor x_i \cdot 2^M \right\rfloor, \quad x_i \in [0; 1[\quad (5.7)$$

$$q_i = \arg_{q_i} \{ \text{UG}(q_i) \leq x_i < \text{UG}(q_i + 1) \} \quad (5.8)$$

Dabei gibt $UG(q_i)$ die untere und $UG(q_i + 1)$ die obere Grenze des Quantisierungsintervalls für das i -te Symbol q_i des Vektors \mathbf{q} an.

Binarisierung (Symbol Mapping)

Die Quantisierungssymbole werden durch die Binarisierung (*Symbol Mapping*) auf mehrere binäre Symbole $q_i^{(1)} \dots q_i^{(M)}$ abgebildet. Dies ist aufgrund der verwendeten binären Slepian-Wolf Codierung notwendig. In der Literatur werden zwei verschiedene Methoden zur Darstellung der Quantisierungssymbole als Binärcode oder als Graycode [Chen u. a. 2008] betrachtet (Tab. 5.1).

Die Simulationsergebnisse (Abb. 5.11) haben gezeigt, dass die Verwendung der Gray-Codierung zu keiner Verbesserung der RD-Performanz führt. Aus diesem Grund wird hier die Binärcodierung verwendet. Die Kombination aus linearer Quantisierung und nachträglicher Zerlegung in binäre Symbole wird oft auch als *nested quantisation* beschrieben.

Die Binarisierung der Quantisierungssymbole wird im Wesentlichen durchgeführt, um eine binäre Slepian-Wolf Codierung zu ermöglichen. Weiterhin kann auch eine symbolbasierte Slepian-Wolf Codierung eingesetzt werden, womit der Binarisierungsschritt entfällt. Wie in [Westerlaken u. a. 2007] für LDPC Codes gezeigt wurde, hat dies keinen Einfluss auf die Leistungsfähigkeit der SW-Codierung.

Des Weiteren bietet die Zerlegung der Quantisierungssymbole in Bitebenen den Vorteil, dass im Verlauf des Decodierungsprozesses der Quantisierungsparameter M durch Übertragung weiterer Bitebenen erhöht werden kann. Damit kann die Rekonstruktionsqualität und die Datenrate angepasst werden.

(a) Binärcode									(b) Graycode								
q_i	0	1	2	3	4	5	6	7	q_i	0	1	2	3	4	5	6	7
$q_i^{(1)}$	0	0	0	0	1	1	1	1	$q_i^{(1)}$	0	0	0	0	1	1	1	1
$q_i^{(2)}$	0	0	1	1	0	0	1	1	$q_i^{(2)}$	0	0	1	1	1	1	0	0
$q_i^{(3)}$	0	1	0	1	0	1	0	1	$q_i^{(3)}$	0	1	1	0	0	1	1	0

Tabelle 5.1.: Binarisierung der Quantisierungssymbole mit Binär- und Graycode

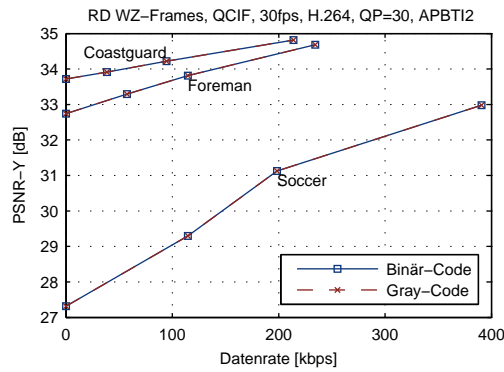


Abbildung 5.11.: Vergleich der RD -Performanz der WZ-Frames unter Verwendung verschiedener Binarisierungsmethoden

Slepian-Wolf Decodierung und Ratenkontrolle

Die Bitebenen der Quantisierungssymbole werden nacheinander unter Kenntnis der Seiteninformation decodiert. Die Datenrate wird mittels Rückkanal kontinuierlich erhöht, bis eine erfolgreiche Decodierung möglich ist (Ratenkontrolle). Die erfolgreiche Decodierung wird, wie aus der Literatur bekannt [Kubasov u. a. 2007a], unter Verwendung eines CRC Codes festgestellt. Ansätze, welche die Nutzung des Rückkanals verringern oder gar vermeiden, werden im Kapitel 7 vorgestellt.

Die weiteren Bitebenen werden unter Kenntnis der Seiteninformation sowie der schon decodierten Bitebenen rekonstruiert. Dies senkt die notwendige Datenrate für Bitebene, die zu einem späteren Zeitpunkt decodiert werden. Wie in [Vatis u. a. 2007] gezeigt wurde, kann durch Anpassung der Decodierreihenfolge die Verteilung der Datenrate zwischen den Bitebenen wesentlich beeinflusst werden. Die Summe der Datenrate über alle Bitebenen bleibt jedoch nahezu konstant.

Rekonstruktion

Die Rekonstruktion bestimmt mit Hilfe des decodierten Quantisierungssymbols \hat{q}_i und der Seiteninformation y_i den rekonstruierten Grauwert \hat{x}_i des Ausgangsbildes. In der Literatur sind im Wesentlichen zwei Verfahren zur Rekonstruktion bekannt:

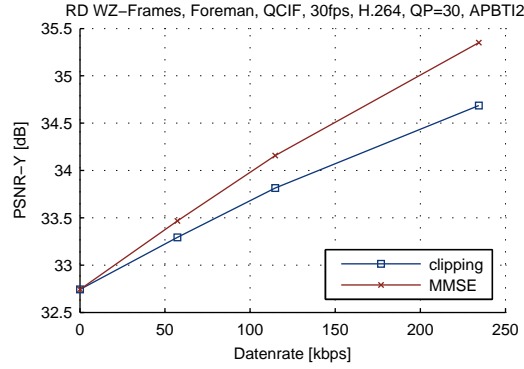


Abbildung 5.12.: Vergleich der Rekonstruktionsmethoden „clipping“ und MMSE

Beschränkung auf das Quantisierungsintervall (clipping) — Bei dieser Art der Rekonstruktion entspricht der Rekonstruktionswert \hat{x}_i der Seiteninformation y_i , wenn y_i innerhalb des Quantisierungsintervalls liegt. Liegt y_i außerhalb des Quantisierungsintervalls, werden die Grenzen des Intervalls als Rekonstruktionswert verwendet (Gl.5.9). Dies ist eine sehr einfache Methode, welche keine Kenntnis über das Kanalmodell erfordert [Aaron u. a. 2002].

$$\hat{x}_i = \begin{cases} \text{UG}(\hat{q}_i) & : y_i \leq \text{UG}(\hat{q}_i) \\ y_i & : \text{UG}(\hat{q}_i) < y_i < \text{UG}(\hat{q}_i + 1) \\ \text{UG}(\hat{q}_i + 1) & : y_i \geq \text{UG}(\hat{q}_i + 1) \end{cases} \quad (5.9)$$

Bestimmung des Erwartungswertes — Ein minimaler quadratischer Fehler (*MMSE*) lässt sich erreichen, indem der Rekonstruktionswert dem Erwartungswert gleichgesetzt wird (Gl. 5.10). Dieser Ansatz wird in [Aaron u. a. 2003b], [Kubasov u. a. 2007b], [Kubasov u. a. 2007b] verwendet und setzt Kenntnis über das Kanalmodell voraus. Die Rekonstruktionsqualität dieser Methode ist höher, womit diese im Weiteren Anwendung findet (Abb. 5.12).

$$\begin{aligned} \hat{x}_i &= \text{E} \{x_i | \hat{q}_i, y_i\} \\ &= \frac{\int_{\text{UG}(\hat{q}_i)}^{\text{UG}(\hat{q}_i+1)} x \cdot p_{X|Y}(x|y_i) dx}{\int_{\text{UG}(\hat{q}_i)}^{\text{UG}(\hat{q}_i+1)} p_{X|Y}(x|y_i) dx} \end{aligned} \quad (5.10)$$

5.2.2. Kanalmodell

Das Kanalmodell bildet die Korrelation zwischen dem Originalbild \mathbf{X} und der Seiteninformation \mathbf{Y} ab. Dieses ist von zentraler Bedeutung für die Verteilte Codierung und beeinflusst die Effizienz der Codierung maßgeblich. Aus diesem Grund soll hier das im Rahmen dieser Arbeit verwendete Modell beschrieben und darauf aufbauend die theoretische Bestimmung der RD-Funktion gezeigt werden.

Abhängigkeit zwischen dem Originalbild und der Seiteninformation

Das Originalbild \mathbf{X} wird als Folge von Symbolen x_i einer kontinuierlichen Zufallsvariablen X beschrieben. Gleiches gilt für die Seiteninformation \mathbf{Y} , wobei die statistische Abhängigkeit durch additive Überlagerung von Rauschen einer Quelle N beschrieben wird (Gl. 5.11, Abb. 5.13). Das Rauschen wird Laplace-verteilt angenommen da sich dies in der Literatur als grobe Näherung etabliert hat.

$$Y = X + N \quad (5.11)$$

$$p_N(n) = \frac{1}{2} \lambda e^{-\lambda|n|} \quad (5.12)$$

$$p_{Y|X}(y_i|x_i) = p_N(y_i - x_i) \quad (5.13)$$

Weiterhin wird angenommen, dass die Zufallsvariablen X und N statistisch unabhängig sind. In dieser Arbeit wird des Weiteren der Modellparameter λ auf Frame-Ebene als bekannt vorausgesetzt.

Bestimmung der intrinsischen Softwerte (LLR) für die Slepian-Wolf Decodierung

Die zweite wichtige Aufgabe des Kanalmodells ist es, die Grundlage zur Bestimmung von Zuverlässigkeitswerten für die einzelnen Bitebenen $q_i^{(b)}$ der Quantisierungssymbole q_i zu liefern. Die Verwendung von Zuverlässigkeitswerten erhöht die Leistungsfähigkeit der Slepian-Wolf Codierung stark, da diese durch Turbocodierung (Abschn. 2.1.2, S. 13) realisiert ist.

Hierzu ist es zunächst notwendig, die Abhängigkeit der Seiteninformation von den Quantisierungssymbolen zu beschreiben, um die intrinsische Information (*Log-Likelihood-Ratio*) zu bestimmen. Abbildung 5.13 zeigt den zuvor beschriebenen Zusammenhang zwischen den Zufallsvariablen $Y = X + N$ beziehungsweise deren Realisierungen $\mathbf{Y} = \mathbf{X} + \mathbf{N}$.

Die Abhängigkeit der Seiteninformation von den Quantisierungssymbolen wird durch $p_{Y|Q}(y|q)$ beschrieben, um diese zu bestimmen, wird zunächst die Abhängigkeit der Originaldaten von den Quantisierungssymbolen beschrieben. Dies wird durch die bedingte Dichte $p_{X|Q}(x|q)$ ausgedrückt, welche sich durch Partitionierung und Normierung der Dichte $p_X(x)$ bestimmen lässt (Gl. 5.14).

$$p_{X|Q}(x|q) = \begin{cases} \frac{p_X(x)}{\int_{\text{UG}(q)}^{\text{UG}(q+1)} p_X(\eta) d\eta} & : \text{UG}(q) \leq x < \text{UG}(q+1) \\ 0 & : \text{sonst} \end{cases} \quad (5.14)$$

Der Übergang zur gesuchten Dichte $p_{Y|Q}(y|q)$ ist durch Faltung mit der Rauschdichte $p_N(n)$ zu erreichen, wenn $Y = X + N$ sowie die statistische Unabhängigkeit zwischen X und N vorausgesetzt wird (Gl. 5.15).

$$p_{Y|Q}(y|q) = p_{X|Q}(y|q) * p_N(y) \quad (5.15)$$

Der gesuchte intrinsische LLR-Wert $L_c(q_i^{(b)})$ für die aktuelle Bitebene b kann nach Gleichung 5.16 bestimmt werden. Es werden nur Quantisierungssymbole berücksichtigt, welche mit den schon decodierten Bitebenen k übereinstimmen $q^{(k)} = \hat{q}_i^{(k)}$. Die Nutzung schon decodierter Bitebenen erhöht die Leistungsfähigkeit der WZ-Codierung wesentlich.

$$L_c(q_i^{(b)}) = \ln \frac{\sum_{q|q^{(b)}=0 \wedge q^{(k)}=\hat{q}_i^{(k)}} p_{Y|Q}(y_i|q)}{\sum_{q|q^{(b)}=1 \wedge q^{(k)}=\hat{q}_i^{(k)}} p_{Y|Q}(y_i|q)} \quad (5.16)$$

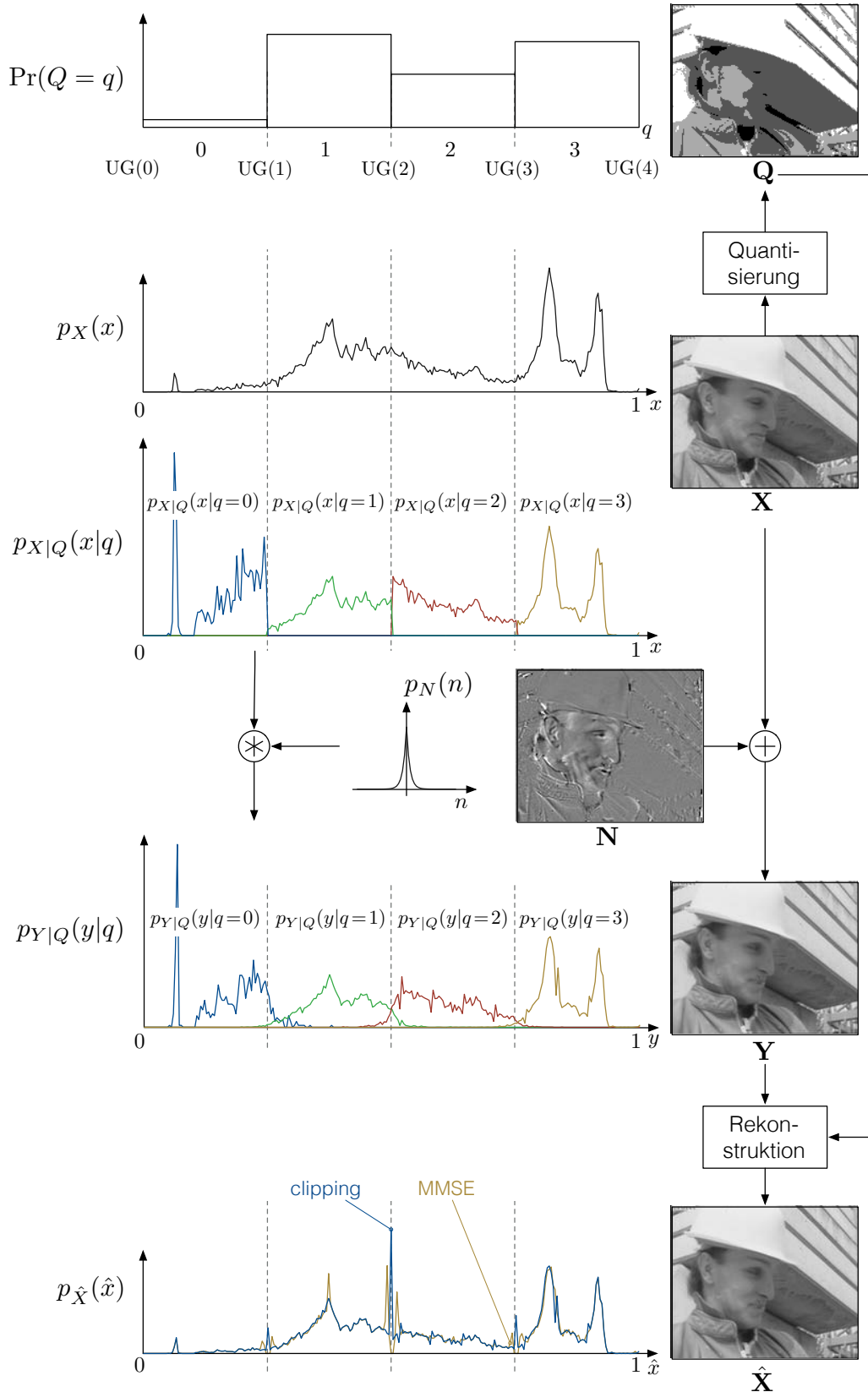


Abbildung 5.13.: Modellierung der Korrelation zwischen Quantisierungssymbol q_i und der Seiteninformation y für die Wyner-Ziv Decodierung, mit $2^M = 4$ Quantisierungsstufen, Foreman, 30 fps, WZ-Frame 85

Diese Herangehensweise ist aus der Trelliscodemodulation (TCM, *trellis-coded modulation*, [Ungerboeck 1987]) bekannt. Dort wurden die Verwendung von Softinformation und die gemeinsame Decodierung vorgeschlagen.

Vereinfachtes Modell

Wird die Dichte $p_X(x)$ als gleichverteilt angenommen und eine lineare Quantisierung verwendet, so kann das Kanalmodell vereinfacht werden (Abb. 5.14). Dies ist oft der Fall, da am Empfänger die Verteilung der Eingangssymbole nicht bekannt ist und als gleichverteilt angenommen wird.

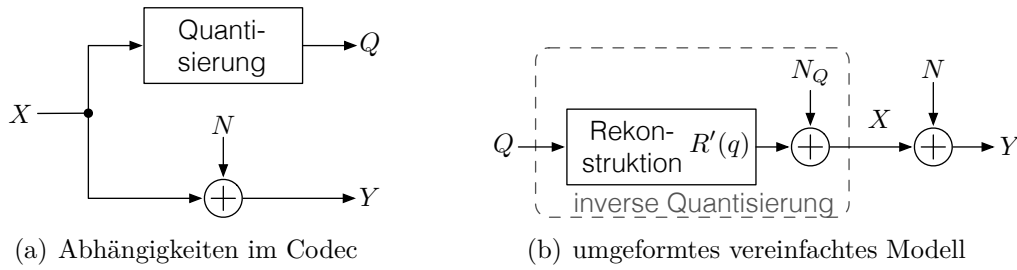


Abbildung 5.14.: Vereinfachte Modellierung der Korrelation zwischen Quantisierungssymbol q_i und der Seiteninformation y_i (Voraussetzung: X gleichverteilt, lineare Quantisierung)

Abbildung 5.14(a) zeigt die Abhängigkeit der Zufallsvariablen Y und Q von X , diese kann, wie in Abbildung 5.14(b) dargestellt, umgeformt werden. Die Umkehrung der Quantisierung wird durch die Rekonstruktion $R'(q)$ und dem zusätzlichen Quantisierungsrauschen N_Q repräsentiert. Diese Darstellung entspricht dem in der Nachrichtentechnik bekannten Aufbau (Abb. 2.4(b)) aus Symbolformer (hier Rekonstruktion, $R'(q)$) und dem Rauscheinfluss N_Q Gl. 5.12 und N Gl. 5.17).

$$p_{N_Q}(n) = \begin{cases} \frac{1}{\Delta} & : |n| < \frac{\Delta}{2} \\ 0 & : \text{sonst} \end{cases} \quad (5.17)$$

$$R'(q) = 2^{-M} \left(q + \frac{1}{2} \right) \quad (5.18)$$

Schlussfolgerungen

In der Beschreibung des Modells wurde angenommen, dass die Zufallsvariablen X und N statistisch unabhängig sind. Wie die Realisierungen \mathbf{X} , \mathbf{N} in Abbildung 5.13 zeigen, ist dies nicht der Fall, was zu einer Verschlechterung der Leistungsfähigkeit der WZ-Codierung führen kann.

Weiterhin wird die örtliche Korrelation der Seiteninformation, des Originalbildes und auch der Quantisierungssymbole nicht berücksichtigt.

Der resultierende Rauscheinfluss setzt sich aus der Qualität der Seiteninformation (Rauschen N) und dem Quantisierungsrauschen N_Q zusammen. Ersteres kann durch verbesserte Verfahren zur Bestimmung der Seiteninformation reduziert werden. Der Einfluss des Quantisierungsrauschens kann hingegen nicht reduziert werden und bleibt unabhängig von den Codierungsparametern erhalten.

5.2.3. Rate-Distortion Funktion und Rekonstruktion

Dieser Abschnitt gibt eine kurze Betrachtung der theoretischen Ableitung der Rate-Distortion Funktion für die verwendete Wyner-Ziv Codierung.

Die Gleichungen 5.19 und 5.20 beschreiben die Rate-Distortion Funktion $R(D)$ theoretisch [Cover u. Thomas 1991][Wyner u. Ziv 1976], wobei $h(\dots)$ die differentielle Entropie darstellt.

$$R = I(X, \hat{X}|Y) = h(X|Y) - h(X|\hat{X}, Y) \quad (5.19)$$

$$\begin{aligned} D &= E\{(X - \hat{X})^2\} \\ &= \int p_X(x) \int p_{\hat{X}|X}(\hat{x}|x)(x - \hat{x})^2 d\hat{x} dx \end{aligned} \quad (5.20)$$

Die Dichte $p_{\hat{X}|X}(\hat{x}|x)$ kann direkt durch Betrachtung der Rekonstruktion gewonnen werden. Bei der Rekonstruktion durch Beschränkung auf das Quantisierungsintervall (*clipping*) entspricht die gesuchte Dichte der schon bekannten Dichte $p_{Y|X}(y|x)$. Werte, die außerhalb des Quantisierungsintervalls liegen, werden auf die Grenzen

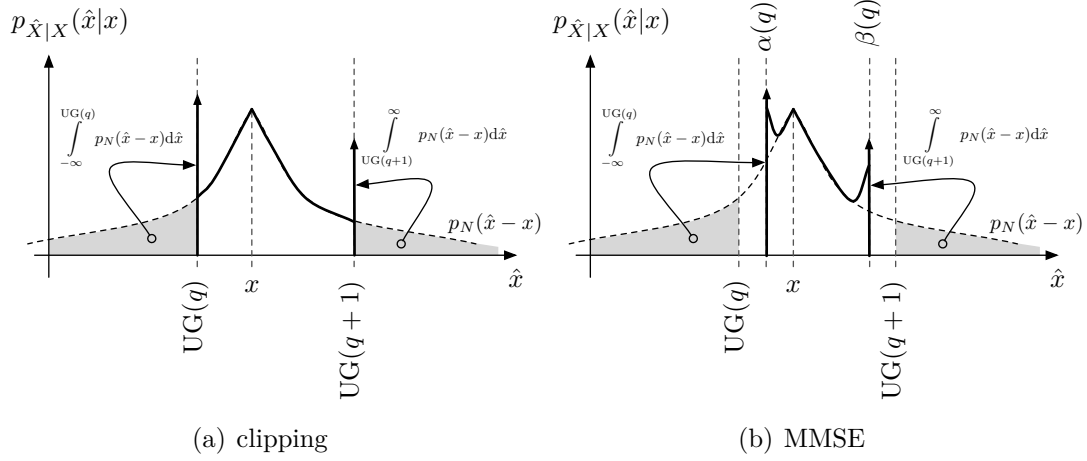


Abbildung 5.15.: Dichte der Rekonstruktionswerte \hat{x} in Abhängigkeit von den Originalgrauwerten x bei Verwendung der Rekonstruktionsmethode „clipping“ bzw. MMSE und des Laplace-Kanalmodells

($UG(q)$ bzw. $UG(q+1)$) abgebildet, wodurch entsprechend zwei Dirac-Impulse in $p_{\hat{X}|X}(\hat{x}|x)$ entstehen (Abb. 5.15(a)).

Für andere Rekonstruktionsmethoden kann die notwendige Dichte $p_{\hat{X}|X}(\hat{x}|x)$ durch Transformation der Dichte $p_{Y|X}$ mit einer nichtlinearen Kennlinie $\hat{x} = f_q(y)$ bestimmt werden [Hänsler 2001, S.135]. Für die MMSE Rekonstruktion kann diese durch Umformung von Gleichung 5.10 gewonnen werden (Gl. 5.24, Abb. 5.16).

$$g_q(y) = \frac{e^{\lambda(UG(q)-y)}(\lambda UG(q) - 1) + e^{\lambda(y-UG(q+1))}(1 + \lambda UG(q+1)) - 2\lambda y}{\lambda(e^{\lambda(UG(q)-y)} + e^{\lambda(y-UG(q+1))}) - 2} \quad (5.21)$$

$$\alpha(q) = g_q(UG(q)) \quad (5.22)$$

$$\beta(q) = g_q(UG(q+1)) \quad (5.23)$$

$$f_q(y) = \begin{cases} \alpha(q) & : y < UG(q) \\ g_q(y) & : UG(q) \leq y \leq UG(q+1) \\ \beta(q) & : y > UG(q+1) \end{cases} \quad (5.24)$$

Die Parameter α und β geben den konstanten Rekonstruktionswert für den Fall an, dass y außerhalb des Quantisierungsintervalls liegt.

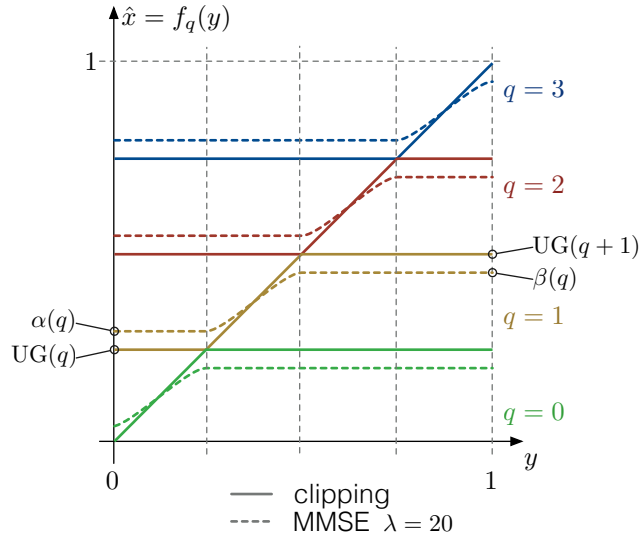


Abbildung 5.16.: Kennlinie $f_q(y)$ der Rekonstruktionsmethoden „clipping“ und MMSE für 4 Quantisierungsintervalle ($M = 2$)

Die Dichte $p_{\hat{X}|X}(\hat{x}|x)$ kann für diesen Fall nicht analytisch bestimmt werden, da die Kennlinie $\hat{x} = f_q(y)$ nicht invertierbar ist. Somit muss eine entsprechende Näherung oder numerische Lösung gewählt werden. Weiterhin konvergiert die Kennlinie für große λ (starke Korrelation) gegen die Kennlinie der Rekonstruktionsmethode „clipping“.

Zwei Erkenntnisse können aus der hier beschriebenen theoretischen Beschreibung der Rate-Distortion Funktion gewonnen werden. Zum einen hängt die RD-Funktion und somit die Leistungsfähigkeit nicht von der verwendeten Binarisierung (*symbol mapping*) ab, da ausschließlich die Art der Quantisierung und Rekonstruktion Einfluss auf die RD-Funktion haben, nicht aber die Repräsentation der Quantisierungssymbole. Dies wurde in Abbildung 5.11 (S. 114) auch empirisch gezeigt.

Zum anderen können durch die verwendete Art der Rekonstruktion Fehler in der Seiteninformation ausschließlich in der Nähe von Quantisierungsintervallgrenzen korrigiert werden (Abb. 5.15, Abb. 5.13). Werte y der Seiteninformation innerhalb des Quantisierungsintervalls werden von den Rekonstruktionsmethoden nur geringfügig oder gar nicht verändert. Werte y außerhalb des Quantisierungsintervalls treten nur auf, wenn starke Störungen auftreten und der Originalwert x dicht an der Intervallgrenze liegt.

5.2.4. Ungleichmäßig verteilte Quellensymbole

Die ungleichmäßige Verteilung der Quellensymbole und deren Einfluss auf die Slepian-Wolf Codierung wurde in Abschnitt 5.1.3 (S. 106) untersucht. Nun soll betrachtet werden, welche Auswirkungen auf die Wyner-Ziv Codierung stattfinden.

Es werden die folgenden vier Ansätze zur Anpassung der Wyner-Ziv Codierung an ungleichmäßig verteilte Eingangssymbole (x_i) betrachtet:

UNI — lineare Quantisierung und keine Berücksichtigung ungleichmäßig verteilter Symbole in der SW Codierung

AIT — lineare Quantisierung und Berücksichtigung der ungleichmäßig verteilten Symbole durch AIT (Übertragung von L_a , Abschn. 5.1.3, S.107) in der SW-Codierung

AIE — lineare Quantisierung und Berücksichtigung der ungleichmäßig verteilten Symbole durch AIE (Schätzung von L_a , Abschn. 5.1.3, S.109) in der SW-Codierung

AQ — adaptive Quantisierung, um gleichmäßig verteilte Quantisierungssymbole zu erhalten

Die Methoden AIT und AIE verlagern die Lösung des Problems auf die Slepian-Wolf Codierung, wobei die adaptive Quantisierung (AQ) eine Lösung direkt im Rahmen der WZ-Codierung bietet. Dabei werden die Quantisierungsintervalle anhand des Histogramms so gewählt, dass gleichverteilte Quantisierungssymbole auftreten (Abb. 5.17).

Die Simulationsergebnisse in Abbildung 5.18 zeigen, dass entgegen den Erwartungen durch die Anwendung von AIT, AIE oder AQ keine Verbesserung der Leistungsfähigkeit zu erreichen ist.

Die gemeinsame Anwendung der Methoden UNI und AIE zeigt eine sehr geringe Verbesserung der Leistungsfähigkeit. Es wird adaptiv die Methode gewählt, welche jeweils die geringste Datenrate zur erfolgreichen Decodierung benötigt. Die adaptive Auswahl erfolgt auf Grundlage des CRC Codes, mit welchem eine erfolgreiche Decodierung festgestellt werden kann.

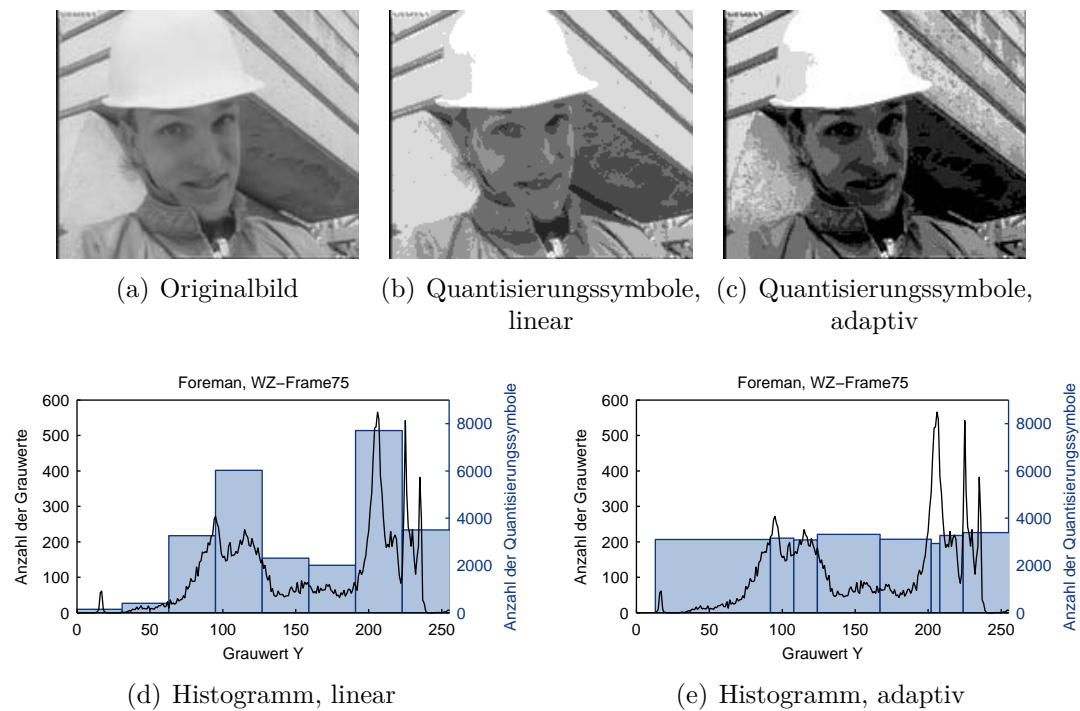


Abbildung 5.17.: Adaptive Quantisierung und gleichverteilte Quantisierungssymbole, Foreman, 30 fps, WZ-Frame 75

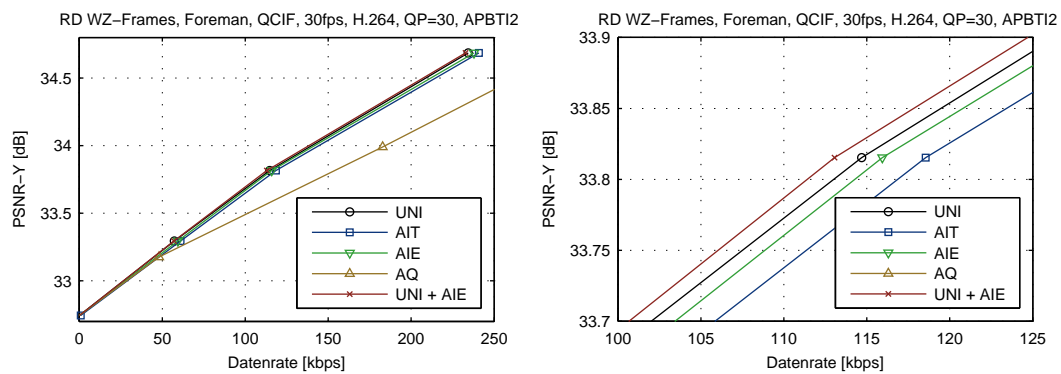


Abbildung 5.18.: Mittlere RD-Performanz der WZ-Frames, Foreman, (rechts) Detailansicht

Abschließend kann festgehalten werden, dass die Kombination aus UNI und AIE den besten Kompromiss in Bezug auf Leistungsfähigkeit, Übertragungskosten und Encoderkomplexität bietet (Tab. 5.2).

	AIT	AIE	AQ	UNI + AIE
RD Performanz	–	–	--	+
Übertragungskosten	–	○	–	○
Encoderkomplexität	--	○	--	○

Tabelle 5.2.: Vergleich der Methoden zur Wyner-Ziv Codierung von ungleichmäßig verteilten Quellsymbolen (– negativ, ○ neutral, + positiv)

Weiterhin wurde in [Li u. a. 2004] die turbocodebasierte SW-Codierung für binäre ungleichmäßig verteilte Quellen betrachtet, jedoch nicht auf die Verteilte Video-codierung angewendet. In [Toto-Zarasoia u. a. 2009] werden *distributed arithmetic codes* (DAC) vorgeschlagen, welche bei starker Korrelation zwischen X und Y eine bessere Leistungsfähigkeit als Turbocodes zeigen.

5.2.5. Fazit

In diesem Abschnitt 5.2 wurde die verlustbehaftete Wyner-Ziv Codierung, basierend auf der verlustlosen Slepian-Wolf Codierung, für Videosequenzen betrachtet. Der grundlegende Aufbau, bestehend aus Pixelauswahl, Quantisierung, Binarisierung, Codierung, Ratenkontrolle und Rekonstruktion, wurde dargelegt.

Die für die Decodierung wichtige Modellbildung wurde beschrieben. Diese erlaubt es, die RD-Funktion (*rate-distortion function*) theoretisch herzuleiten. Des Weiteren konnte damit die Unabhängigkeit der RD-Performanz von der Binarisierung sowohl empirisch als auch theoretisch begründet werden. Weiterhin ist das Modell zwingend erforderlich, um die Softwerte (LLR) für eine leistungsfähige SW-Decodierung zur Verfügung zu stellen.

Für die Codierung von Quellen mit ungleichmäßig verteilten Symbolen wurde, neben den bei der Slepian-Wolf Codierung betrachteten Verfahren AIE und AIT, die adaptive Quantisierung (AQ) vorgeschlagen. Es hat sich jedoch gezeigt, dass diese zu keiner Verbesserung der Leistungsfähigkeit führt, wobei die Kombination aus

Schätzung der a-priori Information (AIE) und der Annahme gleichmäßiger Verteilung (UNI) die besten Ergebnisse liefert.

5.3. Einfluss des Verfahrens zur Codierung der Key-Frames auf das Gesamtsystem

Bisher wurde ausschließlich die RD-Performanz der Wyner-Ziv Frames betrachtet. In dem hier entwickelten Codierungssystem *oDeViCe* werden Key- und Wyner-Ziv Frames abwechselnd codiert. Dadurch beeinflussen die Key-Frames zum einen die RD-Performanz, da diese ein Teil der Bildfolge sind. Des Weiteren beeinflussen diese auch indirekt die Qualität der WZ-Frames, da die Generierung der Seiteninformation stark von diesen abhängt. Dies hat Einfluss auf die Rekonstruktionsqualität und auf die notwendige Datenrate zur erfolgreichen Decodierung der WZ-Frames.

In diesem Abschnitt soll die Leistungsfähigkeit unterschiedlicher Einzelbildcodierungsverfahren gegenübergestellt werden. Weiterhin soll analysiert werden, inwieweit diese Verfahren eine gute Basis zur Generierung der Seiteninformation darstellen.

5.3.1. RD-Performanz der Key-Frame-Codecs

Zur Codierung der Key-Frames werden die im Folgenden kurz vorgestellten Verfahren JPEG, JPEG-2000, H.264intra und HEVCintra betrachtet. Hierbei handelt es sich um Einzelbildcodierungsverfahren oder um Verfahren zur Codierung von Videosequenzen, wobei die zeitliche Prädiktion nicht angewendet wird. Dieser Ansatz wurde gewählt, um die Komplexität des Encoders auf einem geringen Niveau zu belassen.

JPEG — JPEG [Wallace 1992] ist ein weitverbreiteter Standbildcodierer, welcher von der *Joint Photographic Expert Group* erarbeitet wurde und als ISO Standard (JTC1/SC2/WG10) sowie als CCITT (heute ITU-T) Empfehlung veröffentlicht wurde. Im verlustbehafteten Modus wird eine 8×8 DCT eingesetzt, was bei hohen Kompressionsraten zu merklichen Blockartefakten führt.

JPEG2000 — Im Gegensatz zu JPEG wird in JPEG-2000 [Marcellin u. a. 2000] eine Wavelet-Transformation eingesetzt. Dies führt zu Störungen wie *ringing* und Unschärfe (*blur*). Dieser Standard wurde auch von der *Joint Photographic Expert Group* (ISO/IEC 15444, ITU-T Rec. T.800) entwickelt.

H.264 intra — H.264 ist ein leistungsfähiger Videocodec ([Wiegand u. a. 2003] [H.264 2012], JM-18.0, auch MPEG-4/AVC). Dieser wurde vom *Joint Video Team* (JVT) standardisiert. Zur Codierung von Intra-Bildern werden verschiedene blockbasierte Transformationen und Intraprädiktion eingesetzt. Dies führt auch, wie bei JPEG, zur Bildung von Blockartefakten, welche durch ein Deblocking-Filter stark reduzierbar sind.

HEVC intra — HEVC (*high efficiency video coding*) stellt den Nachfolger des H.264 dar und wird durch das *Joint Collaborative Team on Video Coding* (JCT-VC) entwickelt. Wesentliche Erweiterungen zur Codierung von Intra Bildern stellen die erweiterte Intraprädiktion, die größeren Transformationsblöcke und die neuartige Organisationsstruktur (*coding units*) dar. Dieses Codierungsverfahren befindet sich noch in der Entwicklung und soll voraussichtlich Anfang 2013 verabschiedet werden (Stand 6.8.2012). ([Bross u. a. 2012], HM-7.2 [HEVC 2012], [Meur 2011])

Anhand der visuellen Qualität der rekonstruierten Key-Frames (Abb. 5.19) sind zum einen die unterschiedlichen Arten von Störungen zu erkennen, welche zu einer unterschiedlichen subjektiven (wahrgenommenen) Qualität führen. Die RD-Performanz der betrachteten Key-Frame Codierungsverfahren unterscheidet sich stark (Abb. 5.20). So zeigt HEVCintra eine sehr hohe Leistungsfähigkeit, wobei JPEG eine sehr geringe Performanz aufweist. H.264intra und JPEG-2000 zeigen eine ähnliche RD-Performanz.

Unter Berücksichtigung der subjektiven Qualität und der RD-Performanz kommen zur Codierung der Key-Frames nur H.264intra und HEVCintra in Frage.

5.3.2. Seiteninformationsqualität bei unterschiedlichen Key-Frame-Codecs

Neben der RD-Performanz der Key-Frames spielt auch der Einfluss der Störungen (*distortion*) auf die Generierung der Seiteninformation eine wichtige Rolle. Zu

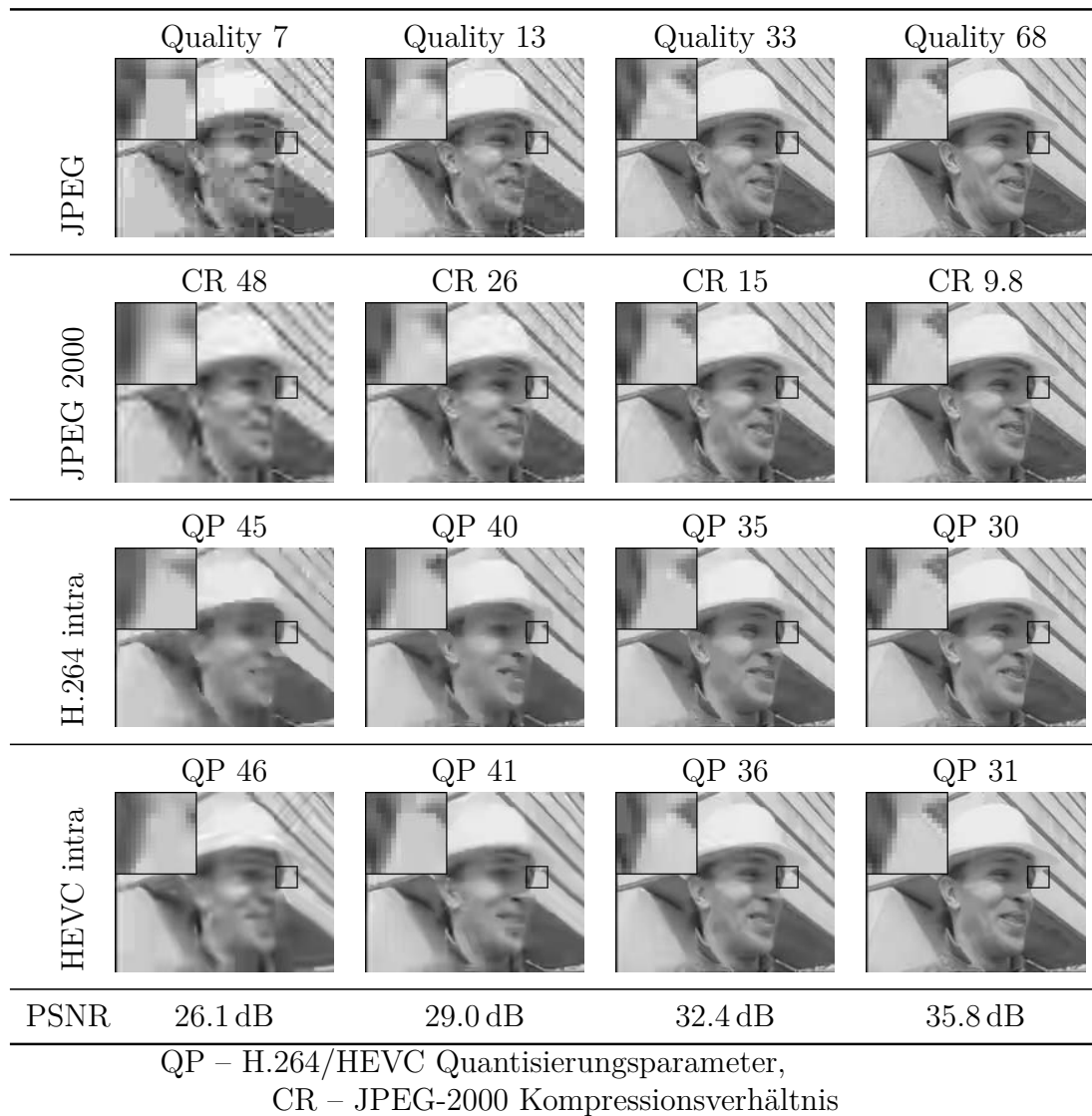


Abbildung 5.19.: Subjektive Qualität der Key-Frames bei gleicher objektiver Qualität PSNR-Y (Foreman, QCIF, Frame 20)

erwarten ist, dass durch die Störungen die Leistungsfähigkeit der Bewegungsschätzung eingeschränkt wird und somit auch die Leistungsfähigkeit der zeitlichen Interpolation beschränkt ist.

Die Simulationsergebnisse zeigen, dass die Qualität der Seiteninformation nur im geringen Maße von dem verwendeten Key-Frame Codec abhängt (Abb. 5.21). Ausschließlich bei Sequenzen mit sehr geringer Bewegung (*Akiyo*, *Hall*, *Mother and Daughter*) zeigt sich ein Vorteil durch Verwendung der Verfahren H.264intra oder HEVCintra.

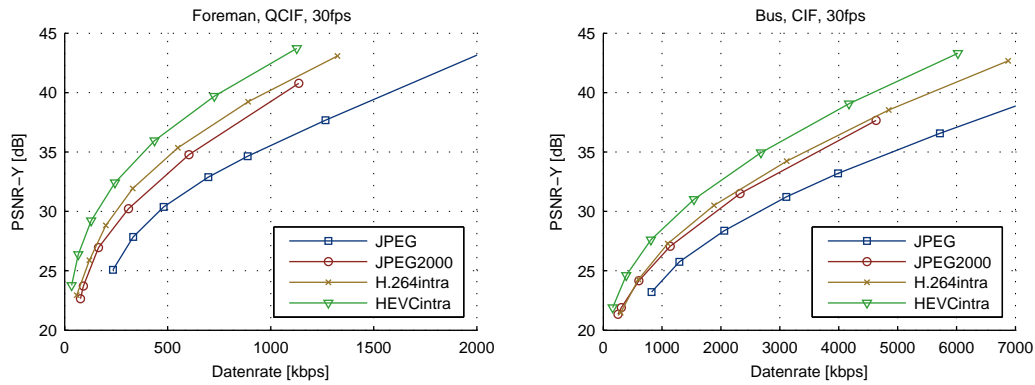


Abbildung 5.20.: RD-Performanz der Key-Frame Codecs für die Sequenz Foreman und Bus

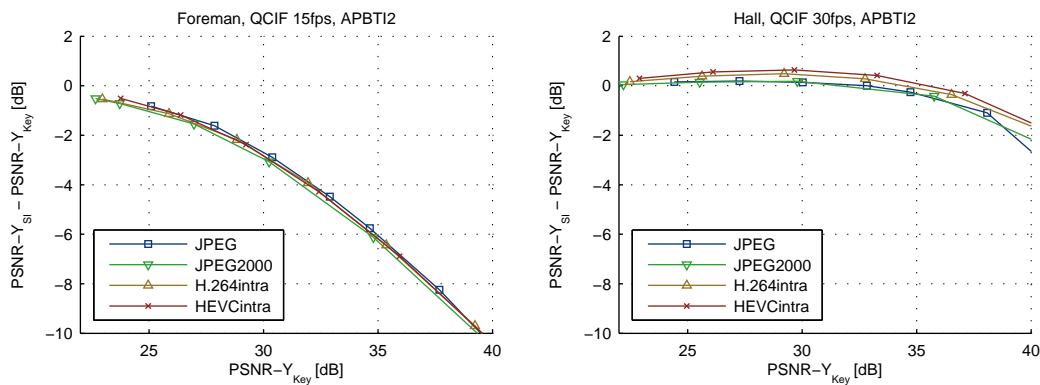


Abbildung 5.21.: Qualität der Seiteninformation im Vergleich zur Qualität der Key Frames

Im Wesentlichen beeinflussen die Kompressionsartefakte die Generierung der Seiteninformation im gleichen Maße. Damit ist aus Sicht der Generierung der Seiteninformation keinem Codierungsverfahren für die Key-Frames Vorrang zu gewähren. Es ist zu erwarten, dass die RD-Performanz der WZ-Frames, je nach Key-Frame Codierungsverfahren, keine wesentlichen Unterschiede zeigt, da diese stark von der Qualität der Seiteninformation abhängt.

5.4. Zusammenfassung

Im vorangegangenen Kapitel 5 wurde die Wyner-Ziv Codierung untersucht. Diese stellt neben der Generierung der Seiteninformation (Kapitel 4) die zweite wichtige

Komponente in einem System zur Verteilten Videocodierung mit geringer Komplexität dar.

Als Basis der verlustbehafteten WZ-Codierung diente die verlustlose Slepian-Wolf Codierung, welche auf Basis von Turbocodes umgesetzt wurde. Hierbei wurde zum einen die Art der Terminierung untersucht und ein Algorithmus zur Erzeugung des Punktierungsschemas vorgeschlagen, welches die sichere Decodierung ermöglicht. Weiterhin wurden Verfahren zum Umgang mit ungleichmäßig verteilten Quellsymbolen vorgeschlagen. Es konnte gezeigt werden, dass Turbocodes für lange Symbolfolgen eine bessere Leistungsfähigkeit als LDPCA Codes aufweisen.

Für die Wyner-Ziv Codierung wurde ein Modell zur leistungsfähigen Codierung vorgeschlagen, welches auch die theoretische Bestimmung der RD-Funktion erlaubt. Weiterhin wurde die ungleichmäßige Verteilung der Quellsymbole betrachtet, wobei sich gezeigt hat, dass nur eine Kombination mehrerer Ansätze zu einer Leistungssteigerung führt.

Abschließend wurde in Hinblick auf die Praxistauglichkeit die Codierung der Key-Frames analysiert und gezeigt, dass die Verfahren H.264intra und HEVCintra eine gute Leistungsfähigkeit des Gesamtsystems erwarten lassen.

Das vorgeschlagene Verfahren zur Codierung der WZ-Frames berücksichtigt die zeitliche Korrelation zwischen den Frames, da die Seiteninformation durch temporale Interpolation gewonnen wird. Die örtliche Korrelation zwischen benachbarten Bildbereichen wird nicht ausgenutzt. Dies könnte weiterführend durch Anwendung der DCT, was die Encoderkomplexität erhöht, berücksichtigt werden. Ein weiterer bekannter Ansatz besteht in der entsprechenden Generierung der Seiteninformation unter Berücksichtigung der örtlichen Korrelation [Tagliasacchi u. a. 2006].

6

Kapitel 6

Leistungsfähigkeit des Gesamtsystems – $\rho DeViCe$

Die Leistungsfähigkeit des entwickelten DVC Codecs $\rho DeViCe$ (Kapitel 3, 4 und 5) wird in diesem Kapitel hinsichtlich der Qualität der Seiteninformation und hinsichtlich der Rate-Distortion Performanz des Gesamtsystems evaluiert.

6.1. Untersuchungsstrategie

Die Leistungsfähigkeit eines DVC Systems hängt wesentlich von der Qualität der Seiteninformation und der Performanz des WZ-Codierers ab.

Die in Kapitel 4 entwickelten Verfahren zur Bestimmung der Seiteninformation sollen hinsichtlich der mittleren Qualität verglichen werden. Dazu wird das arithmetische Mittel des PSNR (*peak signal to noise ratio*) über alle WZ-Frames, aus den Testsequenzen mit 15 fps und 30 fps, herangezogen (Anhang B, S. 213). Weiterhin stehen auch der zeitliche Verlauf des PSNR und die Abhängigkeit des PSNR der Seiteninformation von den Key-Frames im Fokus, da dies wichtig für den subjektiven Eindruck des Betrachters ist.

Der gesamte $\rho DeViCe$ Codec mit Seiteninformationsgenerierung (Kap. 4) und WZ-Codierung (Kap. 5) wird anhand der RD-Performanz (*rate distortion performance*) evaluiert. Hierbei steht sowohl die Wahl des entsprechenden Verfahrens zur

Generierung der Seiteninformation als auch die Wahl des Key-Frame Codec im Mittelpunkt. Neben der mittleren RD-Performanz soll auch der zeitliche Verlauf, das heißt, der PSNR je Frame berücksichtigt werden. Qualitätsschwankungen zwischen benachbarten Bildern haben negative Auswirkung auf die subjektive Qualität.

Die Qualität der Seiteninformation sowie die Rekonstruktionsqualität der Key- und WZ-Frames wird mit der objektiven Metrik PSNR bewertet (Gl. 6.1). Hierfür werden nur die Grauwerte der Bildpunkte berücksichtigt. Die Grauwerte liegen im Wertebereich von 0 bis 1, womit die Normierung auf die maximale Signalleistung entfällt.

$$\text{PSNR}(\mathbf{F}, \hat{\mathbf{F}}) = -10 \log_{10} \left\{ \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N (\mathbf{F}(x, y) - \hat{\mathbf{F}}(x, y))^2 \right\} [\text{dB}] \quad (6.1)$$

6.2. Seiteninformationsqualität

Die Generierung der Seiteninformation ist von zentraler Bedeutung in der Verteilten Videocodierung. Im Kapitel 4 wurden sechs neue Verfahren zur Bestimmung der Seiteninformation, basierend auf der temporalen Interpolation entwickelt. Diese zeigen vor allem in Sequenzen mit schnellen inhomogenen Bewegungen sowie Auf- und Verdeckungen eine sehr unterschiedliche Leistungsfähigkeit (Tab. 6.1).

Als Referenzverfahren werden die aus der Literatur bekannten Methoden BiMESS [Ascenso u. a. 2005], MCTI [Ascenso u. Pereira 2008] und PBTI [Sofke u. a. 2009] herangezogen, um eine Einordnung zu ermöglichen.

6.2.1. Mittlere Qualität der Seiteninformation

Die beste Leistungsfähigkeit unter den betrachteten Verfahren liefert MPBTI, welches für starke inhomogene Bewegungen entwickelt wurde. Unabhängig von der Testsequenz liefert dieses Verfahren die beste bzw. nahezu beste Leistungsfähigkeit

	Bus CIF 15 fps	Bus CIF 30 fps	
BiMESS	20.3 dB	23.7 dB	[Ascenso u. a. 2005]
MCTI	22.7 dB	24.6 dB	[Ascenso u. Pereira 2008]
PBTI	20.4 dB	24.9 dB	[Sofke u. a. 2009]
basicPBTI	19.8 dB	23.6 dB	Abschn. 4.3, S. 47
GTIE	20.5 dB	24.1 dB	Abschn. 4.4.2, S. 57
GpTIE	21.0 dB	25.1 dB	Abschn. 4.4.3, S. 60
MPBTI	24.0 dB	25.8 dB	Abschn. 4.5.2, S. 68
APBTI	20.4 dB	25.9 dB	Abschn. 4.6.1, S. 76
APBTI2	23.4 dB	25.6 dB	Abschn. 4.6.2, S. 78
FPBTI	20.8 dB	25.4 dB	Abschn. 4.7.6, S. 92
Key-Frame	30.5 dB	30.5 dB	

Tabelle 6.1.: Qualität der Seiteninformation und Key-Frames, PSNR-Y, H.264intra Key-Frames, $QP = 35$

(Tab. 6.1, Anhang C.1, S. 217). Weiterhin wird auch durch die Verfahren GpTIE und APBTI2 eine Seiteninformation mit guter Qualität erzeugt.

Die fehlende Adaption an inhomogene Bewegungen führt bei GpTIE zu einer schlechteren Leistungsfähigkeit als bei MPBTI. Weiterhin zeigt das blockbasierte Verfahren GTIE eine geringe Leistungsfähigkeit, da kein dichtes Bewegungsvektorfeld erzeugt wird.

Weiterhin führt die zusätzliche Verwendung der zeitlichen Extrapolation in APBTI2 teilweise zu einer schlechteren Seiteninformationsqualität. Dies ist überraschend, da APBTI2 für Sequenzen mit starken Auf- und Verdeckungen entwickelt wurde. Hier muss demzufolge die Auswahl des Extrapolationsmodus weiter verbessert werden, um eine höhere Leistungsfähigkeit zu erreichen.

Das einfachere adaptive Verfahren APBTI zeigt für hohe Frameraten eine sehr gute Leistungsfähigkeit, wobei die Qualität bei geringeren Frameraten stark einbricht. Dies kann mit dem Fehlen einer Hypothese für die globale Bewegung begründet werden.

Die entwickelte schnelle Umsetzung FPBTI des Ausgangsverfahrens basicPBTI zeigt bei hohen Frameraten eine gesteigerte Leistungsfähigkeit. Diese Verbesserung

kann durch die Ausnutzung der örtlichen Korrelation im Bewegungsvektorfeld erklärt werden, was zu einer zuverlässigeren Bestimmung der Bewegungsvektoren führt.

Eine Kombination aus FPBTI und APBTI2 könnte weiterführend in ein sehr leistungsfähiges und schnelles Verfahren zur Seiteninformationsgenerierung münden, was in weiteren Untersuchungen betrachtet werden sollte.

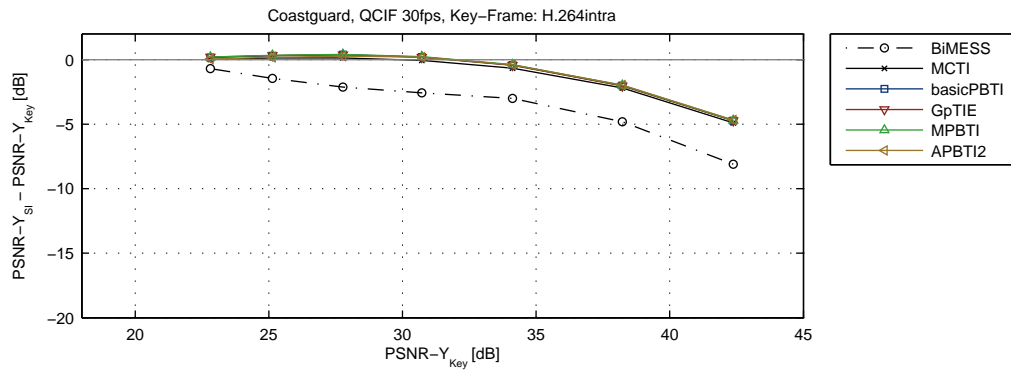
Die Qualität der Seiteninformation steigt, unabhängig von der untersuchten Methode, mit steigender Framerate an, da in diesem Fall auch die Korrelation zwischen den benachbarten Key-Frames größer wird (Tab. 6.1).

6.2.2. Abhängigkeit von der Qualität der Key-Frames

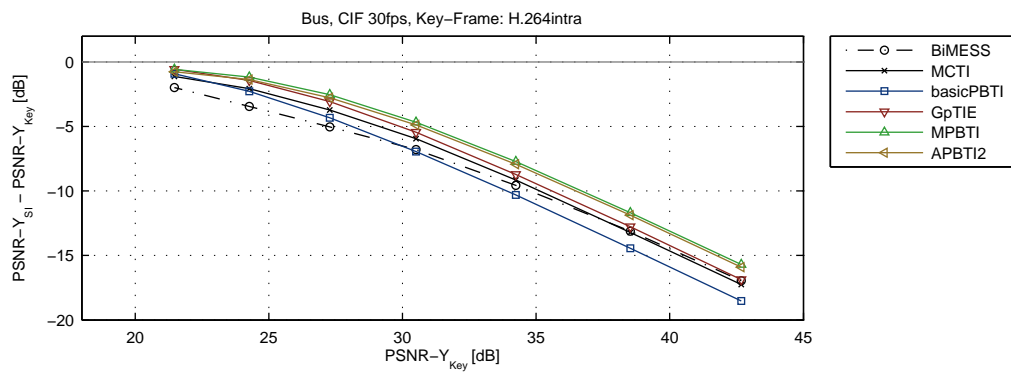
Die Qualität der Seiteninformation ist weiterhin stark von der Rekonstruktionsqualität der Key-Frames abhängig (Abb. 6.1). Im Allgemeinen sinkt die Qualität der Seiteninformation (PSNR-Y) schneller als die Qualität der Key-Frames. Damit wird die Lücke mit sinkendem PSNR der Key-Frames größer, was einen höheren Aufwand und höhere Datenrate für die Decodierung der WZ-Frames bewirkt.

In Ausnahmefällen zeigt die Seiteninformation eine höhere Qualität als die Key-Frames. Dies tritt insbesondere bei Sequenzen mit geringen linearen Bewegungen auf (z.B. *Coastguard*, Abb. 6.1(a)). In diesem Fall ist keine Verbesserung der Qualität der WZ-Frames notwendig, womit auch die Decodierung und Datenübertragung entfällt ($M = 0$).

Dahingegen bildet sich bei Sequenzen mit starker inhomogener Bewegung eine große Diskrepanz zwischen Seiteninformationsqualität und der Qualität der Key-Frames aus. Hierbei ist auch die Wahl des Verfahrens zur Generierung der Seiteninformation entscheidend, da die Qualitätsunterschiede zwischen den Verfahren bis zu 2 dB betragen (z.B. *Bus*, Abb. 6.1(b)).



(a) Coastguard, QCIF, 30 fps



(b) Bus, CIF, 30 fps

Abbildung 6.1.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, PSNR-Y

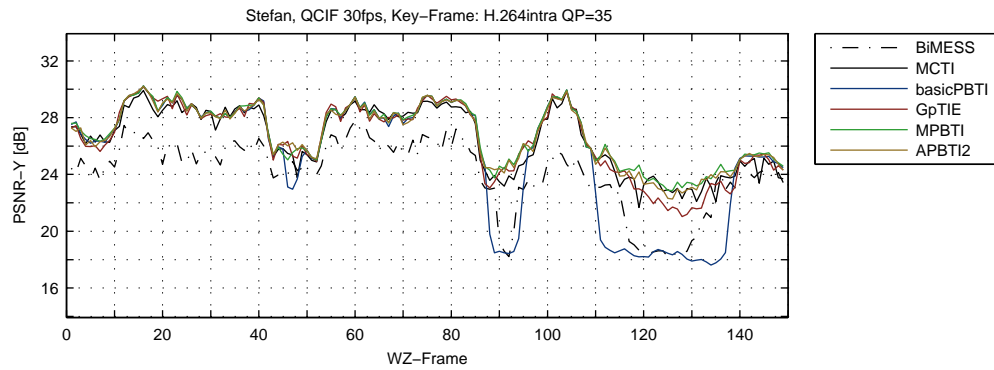


Abbildung 6.2.: Qualität der Seiteninformation je WZ-Frame, **Stefan**, QCIF, 30 fps, Key-Frames: H.264intra, $QP = 35$

6.2.3. Seiteninformationsqualität im Verlauf einer Sequenz

Innerhalb der Testsequenzen unterscheidet sich die Leistungsfähigkeit der Verfahren stark. Besonders in Szenen mit schneller Bewegung (**Stefan**, WZ-Frames 110...140, Abb. 6.2) zeigen die durch die globale Bewegung geführten Verfahren (GpTIE, MPBTI, APBTI2) eine wesentlich höhere Leistungsfähigkeit. Für Szenen mit geringer Bewegung unterscheiden sich die Methoden nur gering.

6.3. RD-Performanz

Die RD-Performanz des entwickelten DVC Codecs $\rho DeViCe$ wird im Vergleich zu konventionellen Videocodecs (H.264, HEVC) untersucht. Dabei soll besonderes Augenmerk auf den Einfluss der Seiteninformationsgenerierung sowie auf den Einfluss des Key-Frame Codecs geworfen werden. Weiterhin sollen die Schwankungen der Rekonstruktionsqualität berücksichtigt werden, da diese großen Einfluss auf die subjektiv wahrgenommene Qualität haben.

6.3.1. Codierungsparameter

Die Sequenzen werden abwechselnd als Key- und WZ-Frames codiert (GOP-Länge 2). Eine Decoder-seitige Ratenkontrolle stellt eine erfolgreiche Decodierung der WZ-Frames sicher, was die Verfügbarkeit eines Rückkanals voraussetzt. Weiterhin

wird angenommen, dass der Parameter des Korrelationsmodells λ für jeden WZ-Frame bekannt ist.

Die Key-Frames werden mittels der Codierungsverfahren H.264intra beziehungsweise HEVCintra codiert (Anhang D.1, S. 245). Die Auswahl der RD-Punkte, welche eine Kombinationen aus Key-Frame Quantisierungsparameter QP und WZ-Frame Quantisierungsparameter M darstellen, wird über den PSNR-Y der Seiteninformation bestimmt (Anhang D.2, S. 246). Je größer die Differenz zwischen PSNR-Y der Seiteninformation und PSNR-Y der rekonstruierten Key-Frames ist, desto größer wird der Quantisierungsparameter der WZ-Frames M gewählt.

6.3.2. Referenzverfahren

Als Vergleich dienen die konventionellen Videocodex H.264 und HEVC, wobei ausschließlich Intra-Bilder codiert werden (H.264intra, HEVCintra). Weiterhin wird auch eine GOP-Struktur mit B-Bildern (I-B-I-B-...) herangezogen (Anhang D.3, S. 246). Die Bestimmung und Übertragung von Bewegungsvektoren erfolgt nicht. Dies verringert die Komplexität des Encoders auf das Niveau des *qDeViCe* Encoders, da so keine Bewegungsschätzung notwendig ist (H.264IBIB). Diese GOP-Struktur wurde so gewählt, dass jedes zweite Frame unter Berücksichtigung seiner benachbarten Frames, wie in *qDeViCe*, codiert wird.

6.3.3. Einfluss des Key-Frame Codec und der Framerate

Die RD-Performanz hängt, wie auch die Qualität der Seiteninformation, stark von der Framerate ab. Bei geringerer Framerate sinkt die RD-Performanz stark (Abb. 6.3). Für eine Sequenz mit mittlerer bis schneller Bewegung (z.B. *Foreman*) wird eine Leistungsfähigkeit ähnlich der eines konventionellen Codecs erreicht. Dies ist für einen pixelbasierten DVC Codec, wie *qDeViCe*, bemerkenswert, da die örtliche Korrelation zwischen den Bildpunkten nicht ausgewertet wird. Die DCT wird, wie in anderen Systemen üblich, nicht zur Dekorrelation eingesetzt (Abschn. 3.2).

Jedes zweite Bild wird als Key-Frame codiert. Damit gehen die Eigenschaften des Key-Frame Codecs zu 50 % in die RD-Performanz ein, wobei zusätzlich die Perfor-

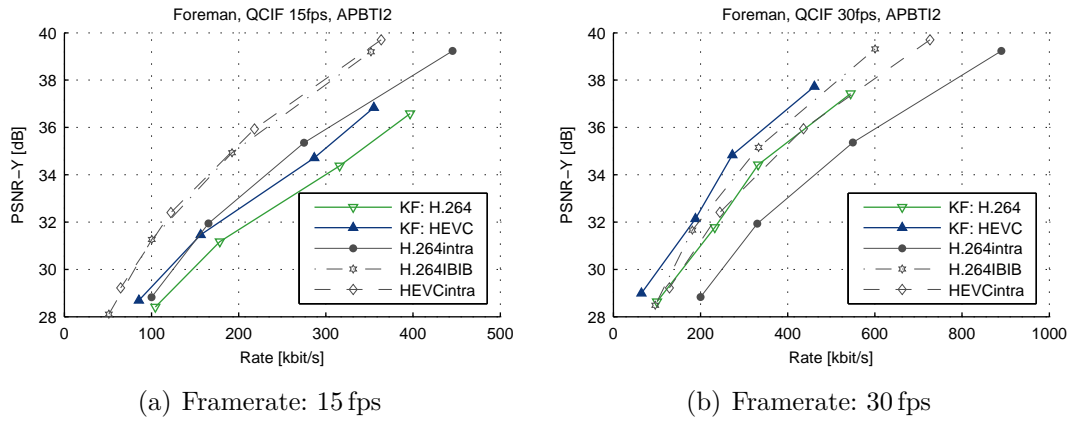


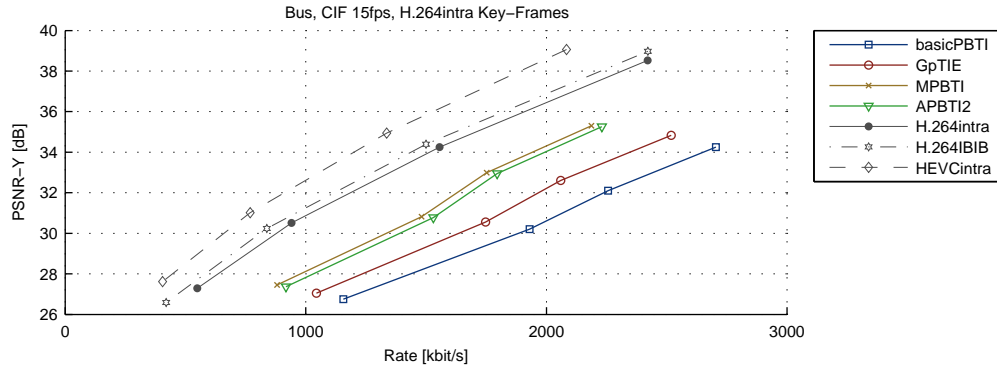
Abbildung 6.3.: RD-Performanz des Gesamtcodecs, **Foreman**, QCIF, Key-Frame Codierung H.264 bzw. HEVC, als Vergleich: konventionellen Codierung mit H.264 und HEVC

manz der WZ-Frames indirekt über die Seiteninformation beeinflusst wird. Durch die Verwendung von HEVCintra zur Codierung der Key-Frames anstatt H.264intra kann die Leistungsfähigkeit des Gesamtsystems weiter gesteigert werden (Abb. 6.3). Jedoch muss auch die Komplexität des Key-Frame Encoders berücksichtigt werden. Soweit bekannt, steht jedoch noch keine vergleichende Untersuchung zwischen HEVC und H.264 in der Literatur zur Verfügung.

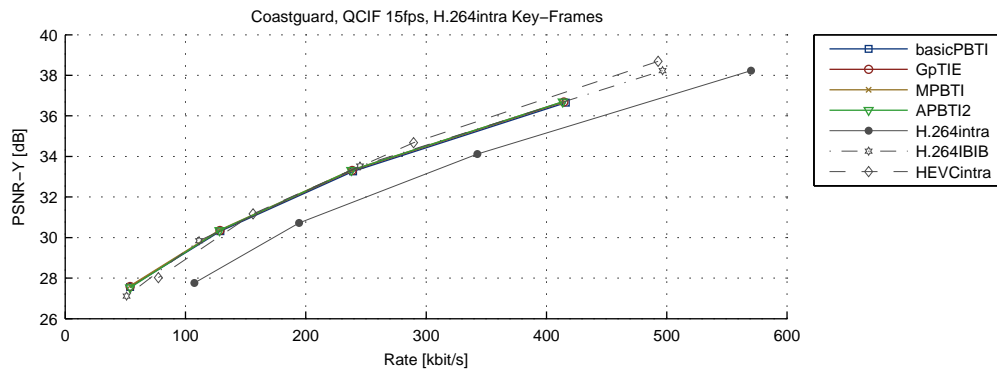
6.3.4. Einfluss der Seiteninformationsgenerierung

Sequenzen mit einer sehr komplexen Bewegung sowie Auf- und Verdeckungen (z.B. **Bus**) stellen eine große Herausforderung für die Verteilte Videocodierung dar. Hier zeigt sich besonders stark der Gewinn durch die entwickelten Verfahren zur Generierung der Seiteninformation (Abb. 6.4(a)). Die beste RD-Performanz wird bei Verwendung des Verfahrens MPBTI erreicht. Dieses Verfahren hatte auch bei der Bewertung der Seiteninformationsqualität sehr gute Ergebnisse geliefert. Die Differenz zum komplexeren Verfahren APBTI2 ist für die Sequenz **Bus** jedoch gering. Die Leistungssteigerung im Vergleich zum Ausgangsverfahren basicPBTI wird durch die mittels globaler Bewegung geführte Bewegungsschätzung (GpTIE) sowie durch die Berücksichtigung inhomogener Bewegungen (MPBTI) erreicht.

Im Vergleich zu den Referenzcodierungsverfahren ist die RD-Performanz des ent-



(a) Bus, CIF, 15 fps



(b) Coastguard, QCIF, 15 fps

Abbildung 6.4.: RD-Performanz des Gesamtcodecs

wickelten DVC Codecs *ρDeViCe* für die Sequenz *Bus* gering, was auf die komplexe Bewegungsstruktur dieser Testsequenz zurückzuführen ist.

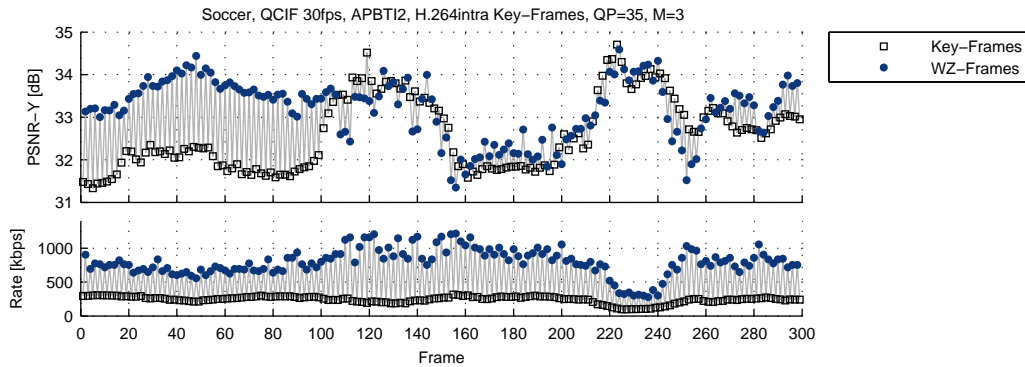
Für Sequenzen mit einfacheren Bewegungsstrukturen (z.B. *Coastguard*) wird eine, im Vergleich zu den Referenzverfahren, gute Leistungsfähigkeit erreicht (Abb. 6.4(b)).

6.3.5. Schwankungen der Rekonstruktionsqualität

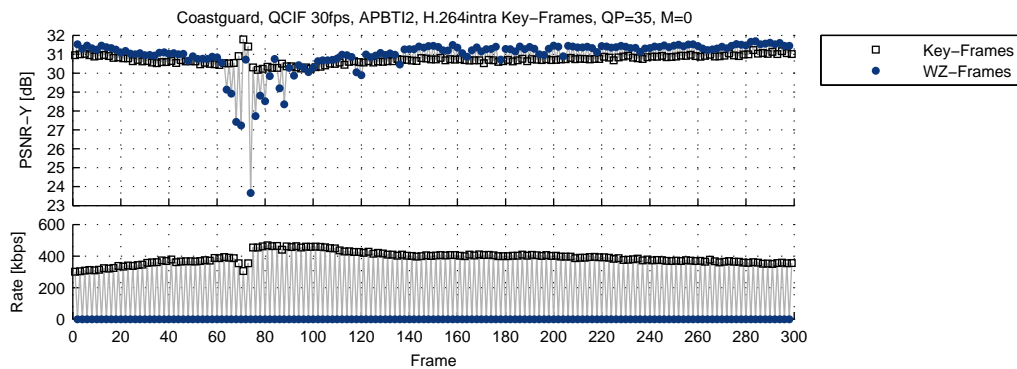
Die vom Betrachter wahrgenommene Qualität einer Videosequenz hängt stark von den Qualitätsschwankungen zwischen den aufeinanderfolgenden Bildern ab. Je RD-Punkt werden die Quantisierungsparameter für Key- und WZ-Frames für die gesamte Sequenz gewählt.

Ändert sich die Charakteristik der Sequenz, so ändert sich auch das Verhältnis zwischen der Rekonstruktionsqualität der Key- und WZ-Frames (z.B. *Soccer*, ab

Frame 100, Abb. 6.5(a)). In diesem Fall könnten die Quantisierungsparameter angepasst werden, was die RD-Performanz weiter verbessern würde.



(a) Soccer



(b) Coastguard

Abbildung 6.5.: Qualität und Datenrate der Key- und WZ-Frames, QCIF, 30 fps

Die kurzfristigen Schwankungen des PSNR zwischen den Frames betragen je nach Sequenz bis zu ± 3 dB, was einen starken Einfluss auf die subjektive Qualität hat. Hier kann eine Wahl der Codierungsparameter je Frame Abhilfe schaffen.

Die Datenrate der WZ-Frames steigt bei geringerer zeitlicher Korrelation stark an, was aus dem Vergleich der Sequenzen *Soccer* und *Coastguard* zu erkennen ist (Abb. 6.5(a) und 6.5(b)). Zum einen muss ein höherer Quantisierungsparameter für die WZ-Frames gewählt werden, um die Lücke zwischen PSNR der Key- und WZ-Frames zu schließen. Zum anderen muss für die WZ-Decodierung eine wesentlich höhere Datenrate aufgewendet werden, um bei schlechterer Qualität der Seiteninformation eine erfolgreiche Decodierung zu erreichen.

Damit erhöht sich die Datenrate der WZ-Frames stark und kann, für Sequenzen

mit komplexen Bewegungen, größer als die Datenrate der Key-Frames werden. Im Gegensatz dazu tritt bei Sequenzen mit einfachen Bewegungsstrukturen der Fall ein, dass keine WZ-Decodierung durchgeführt wird und die Seiteninformation die rekonstruierten WZ-Frames ersetzt. In diesem Fall werden keine Daten für die WZ-Frames übertragen (Abb. 6.5(b)).

6.4. Zusammenfassung

Die Bewertung der Leistungsfähigkeit des entwickelten Verteilten Videocodierungssystems *ρ DeViCe* hat gezeigt, dass für Sequenzen mit hoher zeitlicher Korrelation eine gute RD-Performanz, im Vergleich zu den Referenzverfahren, erreicht wird. Bei sehr starker zeitlicher Korrelation (z.B. **Coastguard**) wurde keine Codierung der WZ-Frames vorgenommen und die Seiteninformation direkt als Ausgabebild verwendet. Da die Seiteninformation einen höheren PSNR als die Key-Frames zeigte, konnte so eine Verbesserung der Leistungsfähigkeit erreicht werden.

Anzumerken ist, dass der entwickelte Codec *ρ DeViCe* die örtliche Korrelation aufgrund geforderter niedriger Encoderkomplexität nicht auswertet. Die Verwendung der DCT könnte die Leistungsfähigkeit weiter steigern.

Für Sequenzen mit komplexen Bewegungsstrukturen konnte, insbesondere durch die neu entwickelten Verfahren zur Generierung der Seiteninformation, eine Verbesserung der Seiteninformationsqualität sowie eine wesentliche Steigerung der RD-Performanz erreicht werden. Dies führte dazu, dass der entwickelte Codec für hohe Frameraten eine mit den Referenzverfahren vergleichbare Leistungsfähigkeit zeigt.

Entgegen den Erwartungen zeigte das in dieser Arbeit am weitesten entwickelte Verfahren zur Generierung der Seiteninformation APBTI2, welches globale, inhomogene als auch Auf- und Verdeckungen berücksichtigt, nicht die beste Leistungsfähigkeit. In Bezug auf die Qualität der Seiteninformation sowie in Bezug auf die RD-Performanz wurden die besten Ergebnisse in der Mehrzahl der Fälle mit dem Verfahren MPBTI erreicht (Abschn. 6.3.4).

7

Kapitel 7

Rückkanal in der Verteilten Videocodierung

Ein großes Problem in der Verteilten Codierung stellt die Kommunikation zwischen Encoder und Decoder dar. In diesem Kapitel werden die Rolle des Rückkanals analysiert und dadurch auftretende Probleme beschrieben. Weiterhin werden aus der Literatur bekannte Lösungsansätze klassifiziert und darüber hinaus der vorgeschlagene Ansatz der flexiblen Decodierung vorgestellt.

In den vorangegangenen Kapiteln wird das Verteilte Videocodierungssystem *qDeViCe* mit geringer Encoderkomplexität entworfen. Dazu wurden die beiden wesentlichen Komponenten „Generierung der Seiteninformation“ (Kap. 4) und die „Wyner-Ziv Codierung“ (Kap. 5) entwickelt und ihre Leistungsfähigkeit evaluiert (Kap. 6). Als Basis für die Kompression einer Videosequenz wurde die verlustlose Slepian-Wolf Codierung herangezogen. Die Ratenkontrolle stellt hier eine besondere Problematik dar, da nur der Decoder Wissen über die notwendige Datenrate besitzt, der Encoder jedoch exakt die notwendige Datenmenge übertragen muss. Dieses Problem wurde im entwickelten Codec *qDeViCe*, wie auch in der Literatur üblich, durch Verwendung eines Rückkanals vom Decoder zum Encoder gelöst.

Die Verwendung eines Rückkanals ist für viele Anwendungsgebiete nicht praktikabel, da dieser oftmals nicht zur Verfügung steht. Aus diesem Grund wird in diesem Kapitel die Rolle des Rückkanals analysiert. Die auftretenden Probleme

werden charakterisiert. Weiterhin werden Lösungsansätze für die einzelnen Problembereiche vorgeschlagen. Dieses Kapitel ist mehr als erweiterter Ausblick zu verstehen. Die Entwicklung kompletter Lösungen wird nicht angestrebt. In der Literatur gibt es, so weit bekannt, keine umfassende Lösung zum Problem des Rückkanals.

7.1. Das Problem Rückkanal

Die Nutzung des Rückkanals ist notwendig, wenn eine Decoder-seitige Ratenkontrolle verwendet wird. Hierbei fordert der Decoder weitere Daten vom Encoder an, bis eine erfolgreiche Decodierung möglich ist. Systeme mit Rückkanal können nur in bestimmten Anwendungsszenarien eingesetzt werden, da eine sofortige Decodierung durchgeführt werden muss. Dies umfasst zum Beispiel Anwendungen, wie die Live-Übertragung einer Videosequenz von einem oder mehreren mobilen Endgeräten.

In Anwendungsszenarien, wie zum Beispiel der Speicherung einer Videosequenz, steht kein Rückkanal zur Verfügung, da zum Zeitpunkt der Decodierung der Encodierungsprozess bereits abgeschlossen ist. Hiermit ist das Anfordern weiterer Daten nicht möglich und somit ist eine erfolgreiche Decodierung nicht sichergestellt. Dies führt bei einer fehlerhaften Decodierung zu einer sehr geringen RD-Performanz. Weiterhin verursachen die wiederholten Decodierungsversuche und das Anfordern weiterer Daten eine hohe Latenz, welche zum einen von der Leistungsfähigkeit des Decoders und zum anderen von der Latenz des Kommunikationskanals abhängt.

Decodierversagen tritt ein, wenn eine zu geringe Datenrate übertragen wurde. Die notwendige Datenrate wird wesentlich durch das verwendete Quantisierungsschema bestimmt, welches zu Beginn der Codierung festgelegt wird. Dieser Parameter kann später nicht angepasst werden, um eine erfolgreiche Decodierung und hohe Rekonstruktionsqualität zu ermöglichen. Die Notwendigkeit des Rückkanals kann durch die Umsetzung der WZ-Codierung durch Quantisierung und SW-Codierung begründet werden (Abb. 5.8). Damit wird zu Beginn der Codierung ein RD-Punkt fest ausgewählt, wodurch der Decoder nicht mehr flexibel auf die übertragene Da-

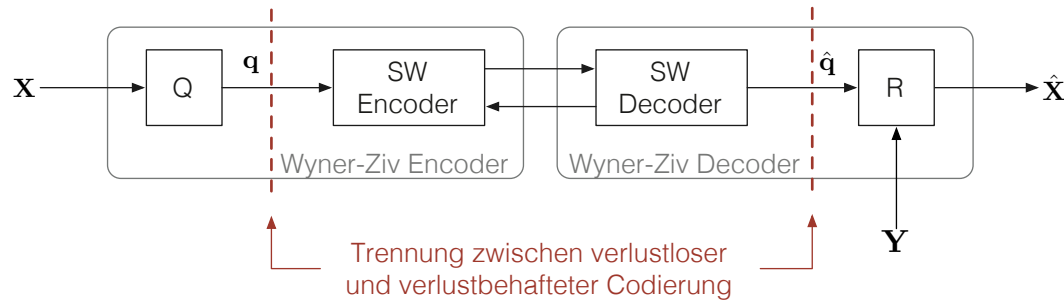


Abbildung 7.1.: Trennung zwischen verlustloser und verlustbehafteter Verteilter Codierung

tenmenge reagieren kann. Die Rekonstruktionsqualität skaliert somit nicht mit der übertragenen Datenmenge.

Zusammenfassend werden durch die Trennung von verlustbehafteter und verlustloser Codierung (Abb. 7.1) zusätzliche Randbedingungen in das DVC-System eingeführt, welche das Problem der Ratenkontrolle verschärfen.

Die Rekonstruktionsqualität in DVC-Systemen nimmt nur dann zu, wenn eine erfolgreiche SW-Decodierung erfolgt ist. Damit wird mit steigender Datenrate die Rekonstruktionsqualität nicht erhöht, solange nicht die notwendige Datenrate für eine erfolgreiche Decodierung erreicht wurde. Zusätzlich übertragene Daten erhöhen die Rekonstruktionsqualität nicht weiter.

Ziel ist es, ein Codierungssystem zu entwerfen, bei dem die Rekonstruktionsqualität kontinuierlich mit steigender Datenrate zunimmt.

7.2. Lösungsansätze

Die einfache Decoder-seitige Ratenkontrolle setzt die volle Nutzung eines Rückkanals voraus. Nach jedem erfolglosen Decodierungsversuch wird ein neues Datenpaket angefordert und ein neuer Versuch gestartet. In der Literatur sind weiterhin Methoden bekannt, um die Nutzung des Rückkanals zu reduzieren [Slowack u. a. 2012a, b]. Weiterhin kann das Ratenkontrollproblem auch durch eine Encoder-seitige Ratenkontrolle gelöst werden [Brites u. Pereira 2007, 2011], womit die Kommunikation vom Decoder zum Encoder entfällt (Abb. 7.2).

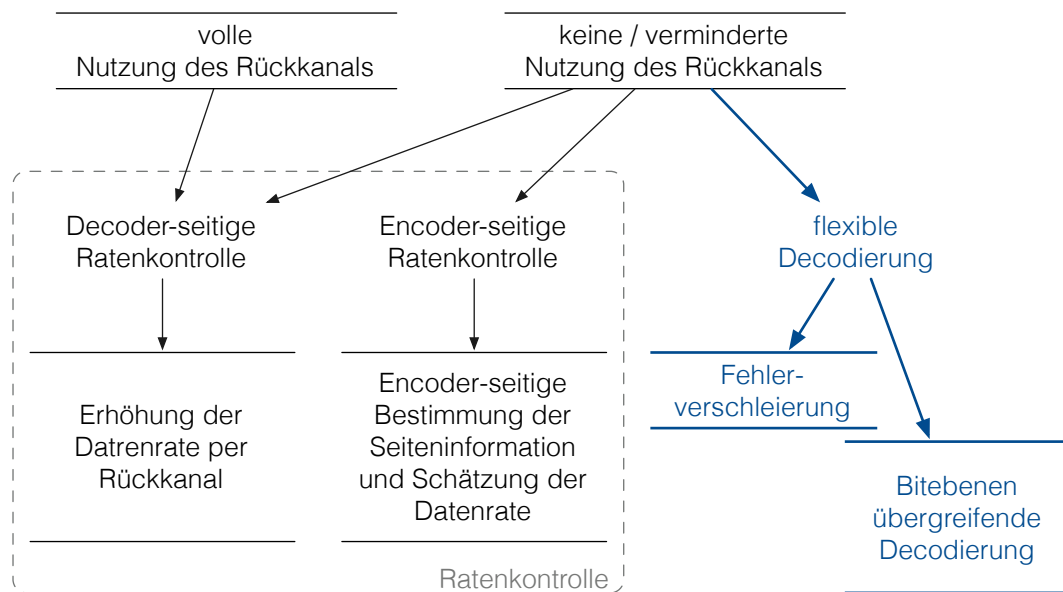


Abbildung 7.2.: Einteilung in Systeme mit und ohne Nutzung des Rückkanals und Zusammenhang zur Ratenkontrolle

Ziel der Encoder- oder Decoder-seitigen Ratenkontrolle ist es, die notwendige Datenrate zur erfolgreichen Decodierung möglichst genau zu bestimmen. Neben diesen Ansätzen aus dem Bereich der Ratenkontrolle wird in dieser Arbeit die flexible Decodierung vorgeschlagen (Abb. 7.2). Die grundlegende Herangehensweise besteht darin, übertragene und somit am Decoder verfügbare Daten zu nutzen, auch wenn diese nicht zu einer erfolgreichen Decodierung führen. Ziel der flexiblen Decodierung ist es somit, dem Decoder mehr Möglichkeiten zu verschaffen, um mithilfe der übertragenen Daten die Rekonstruktionsqualität zu verbessern.

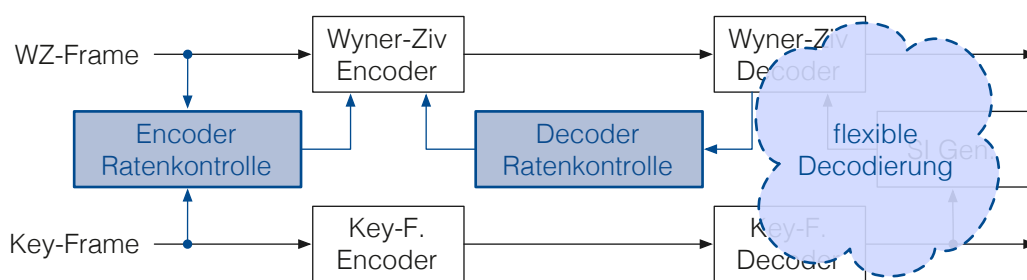


Abbildung 7.3.: Einfaches DVC-System, erweitert um Encoder- und Decoder-seitiger Ratenkontrolle sowie flexibler Decodierung

Die folgende Liste gibt einen Überblick über die Lösungsansätze zur verminderten Nutzung des Rückkanals. Weiterhin zeigt Abbildung 7.3 die schematische Anord-

nung dieser Verfahren in einem Verteilten Codierungssystem.

Decoder-seitige Ratenkontrolle — Bei der Decoder-seitigen Ratenkontrolle handelt es sich um eine einfache Form der Ratenkontrolle, indem nach jedem erfolglosen Decodierversuch ein weiteres Datenpaket angefordert wird. Darüber hinaus gibt es Ansätze, die notwendige Datenrate am Decoder abzuschätzen und somit die Anzahl der Rückfragen zu verringern [Slowack u. a. 2012b, a]. Somit wird auch die Latenz verringert und die Nutzung des Rückkanals reduziert.

Encoder-seitige Ratenkontrolle — Die Schätzung der notwendigen Datenrate für eine erfolgreiche Decodierung am Encoder macht die Nutzung des Rückkanals überflüssig. Das Problem besteht darin, eine exakte Schätzung der Datenrate zu erreichen, wobei die Seiteninformation und somit das Maß der Korrelation zwischen Originalbild X und Seiteninformation Y nicht bekannt ist. Aufgrund der zu erhaltenden geringen Komplexität des Encoders, wird hier meist eine vereinfachte Bestimmung der Seiteninformation durch Mittelung der Nachbarbilder [Priziment u. Malah 2008][Kodavalla u. Mohan 2011][Badem u. a. 2010][Yaacoub u. Farah 2009][Morbee u. a. 2007] oder durch stark vereinfachte Bewegungsschätzung und -kompensation [Brites u. Pereira 2007, 2011] durchgeführt. Stärkere Bewegungen können so auch berücksichtigt werden. Weiterführend wird anhand eines Modells die Datenrate bestimmt. Weitere Ansätze verwenden Entscheidungsbäume, um die Datenrate zu bestimmen (*machine learning*, [Martinez u. a. 2008a, b]). Wird die Datenrate nicht korrekt bestimmt, so tritt eine fehlerhafte Decodierung auf oder die Datenrate ist höher als notwendig. In beiden Fällen wird die RD-Performanz verschlechtert.

Flexible Decodierung — Im Gegensatz zur Decoder- und Encoder-seitigen Ratenkontrolle wird bei der vorgeschlagenen flexiblen Decodierung die Datenrate nicht beeinflusst. Hier ist es das Ziel, durch geschickte Verteilung der Information eine erfolgreiche Decodierung [Hänsel u. Müller 2009c] oder durch eine Fehlerverschleierung eine Verbesserung der Rekonstruktionsqualität [Hänsel u. Müller 2009a, b, 2010] zu erreichen.

Multiple Description — Weitere DVC-Systeme mit verminderter Nutzung des Rückkanals basieren auf *multi description coding* (MDC, [Kuo u. a. 2012]) und

erfordern eine Neustrukturierung des Codecs. Der in [Adikari u. a. 2007] vorgestellte Ansatz führt die Decodierung parallel mit mehreren WZ-Decodern durch, was zu einer Erhöhung der Rekonstruktionsqualität führt.

In dieser Arbeit liegt das Augenmerk auf DVC-Systemen mit geringer Encoderkomplexität. In anderen Anwendungsszenarien, welche die Flexibilität und Driftfreiheit der Verteilten Videocodierung nutzen, können testweise Encodierung und Decodierung durchgeführt werden, um so die notwendige Datenrate zu bestimmen. Dies setzt jedoch eine hohe Rechenleistung des Encoders und die Verfügbarkeit der Seiteninformation voraus, womit dies für Encoder mit geringer Komplexität nicht möglich ist. Das Rückkanalproblem tritt im Wesentlichen bei Encodern mit geringer Komplexität oder bei der Codierung mit örtlich getrennten Encodern auf.

7.3. Anwendungsszenarien für die flexible Decodierung

Bevor auf die einzelnen vorgeschlagenen Verfahren zur flexiblen Decodierung eingegangen wird, sollen hier die entsprechenden Szenarien beschrieben werden. Alle Szenarien beschreiben den Fall, dass eine erfolgreiche Decodierung mit der herkömmlichen SW-Decodierung, aufgrund einer zu geringeren Datenmenge, nicht möglich ist. Der Fall einer zu hohen übertragenen Datenmenge wird nicht behandelt, da hier eine herkömmliche SW-Decodierung und Rekonstruktion durchgeführt werden kann, was zu einem sehr guten Rekonstruktionsergebnis führt.

Die herkömmliche Slepian-Wolf Decodierung verarbeitet die Bitebenen $q^{(b)}$ eines Quantisierungssymbols q nacheinander. Die Reihenfolge der Decodierung hat wesentlichen Einfluss auf die zur erfolgreichen Decodierung der Bitebene notwendigen Datenrate $R^{(b)}$. Diese wird größer, je früher in der Reihenfolge eine Bitebene decodiert wird. Für eine erfolgreiche Decodierung von zum Beispiel zwei Bitebenen müssen folgende Bedingungen erfüllt sein:

- Für eine Bitebene b muss die Datenrate $R_{max}^{(b)}$ erreicht sein, um dieses als erstes zu decodieren. $\exists b, R^{(b)} \geq R_{max}^{(b)}$

- Für die andere Bitebene b' muss die minimal notwendige Datenrate $R^{(b')} \geq R_{min}^{(b')}$ erreicht sein um eine erfolgreiche Decodierung bei Kenntnis der anderen Bitebene, zu ermöglichen.
- Die Summe der übertragenen Datenrate muss größer als die insgesamt zur erfolgreichen Decodierung notwendigen Datenrate sein, $\sum_k R^{(k)} \geq R_{suc}$.

Abbildung 7.4 zeigt eine mögliche Datenverteilung zwischen den Bitebenen bei herkömmlicher Decodierung, wobei die Bitebene 1 als Erstes decodiert wird. Diese Randbedingungen lassen sich auch auf 3 und mehr Bitebenen erweitern, womit weitere Grenzen für die minimale Datenrate, je nach Reihenfolge, eingeführt werden müssen.

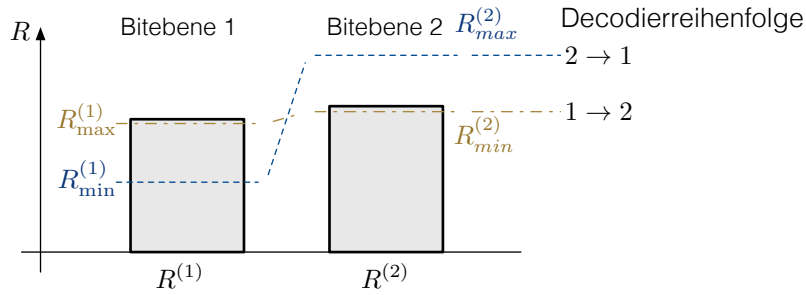


Abbildung 7.4.: Verteilung der Datenrate bei der Bitebenen-weisen Decodierung, zuerst Bitebene 1, dann Bitebene 2 ($1 \rightarrow 2$), $M = 2$

Im Folgenden werden drei Szenarien beschrieben, für welche die flexible Decodierung Lösungsansätze bietet.

Szenario 1 — Die Summendatenrate wurde erreicht $\sum_k R^{(k)} \geq R_{suc}$, jedoch wurde für keine Bitebene die notwendige Datenrate erreicht, um diese als Erstes zu decodieren $R^{(k)} < R_{max}^{(k)}, \forall k$ (Abb. 7.5(a)). Da die herkömmliche Decodierung aufgrund der ungünstigen Verteilung der Datenmenge nicht möglich ist, wird zur Lösung des Problems die Bitebenen übergreifende Decodierung vorgeschlagen, welche alle Bitebenen parallel decodiert (Abschn. 7.4).

Szenario 2 — Die maximale Datenrate für eine Bitebene wurde erreicht $\exists b, R^{(b)} \geq R_{max}^{(b)}$, die Summendatenrate ist jedoch nicht ausreichend zur erfolgreichen Decodierung aller Bitebenen $\sum_k R^{(k)} < R_{suc}$ (Abb. 7.5(b)). In diesem Fall können die Bitebenen, soweit möglich, mittels der herkömmlicher SW-Decodierung

decodiert werden. Die erfolgreich decodierten Bitebene können zur Rekonstruktion verwendet werden. Anschließend kann das Rekonstruktionsergebnis durch die Nutzung von Zuverlässigkeitsinformationen (Abschn. 7.5) oder durch eine entsprechende Fehlerverschleierung (Abschn. 7.6) verbessert werden.

Szenario 3 — Die notwendige Summendatenrate $\sum_k R^{(k)} < R_{suc}$ und die notwendige Datenrate zur Decodierung der einzelnen Bitebenen $R^{(k)} < R_{max}^{(k)}, \forall k$ wurden nicht erreicht (Abb. 7.5(c)). In diesem Fall ist keine herkömmliche oder Bitebenen übergreifende Decodierung möglich, somit wird hier eine Fehlerverschleierung vorgeschlagen (Abschn. 7.6).

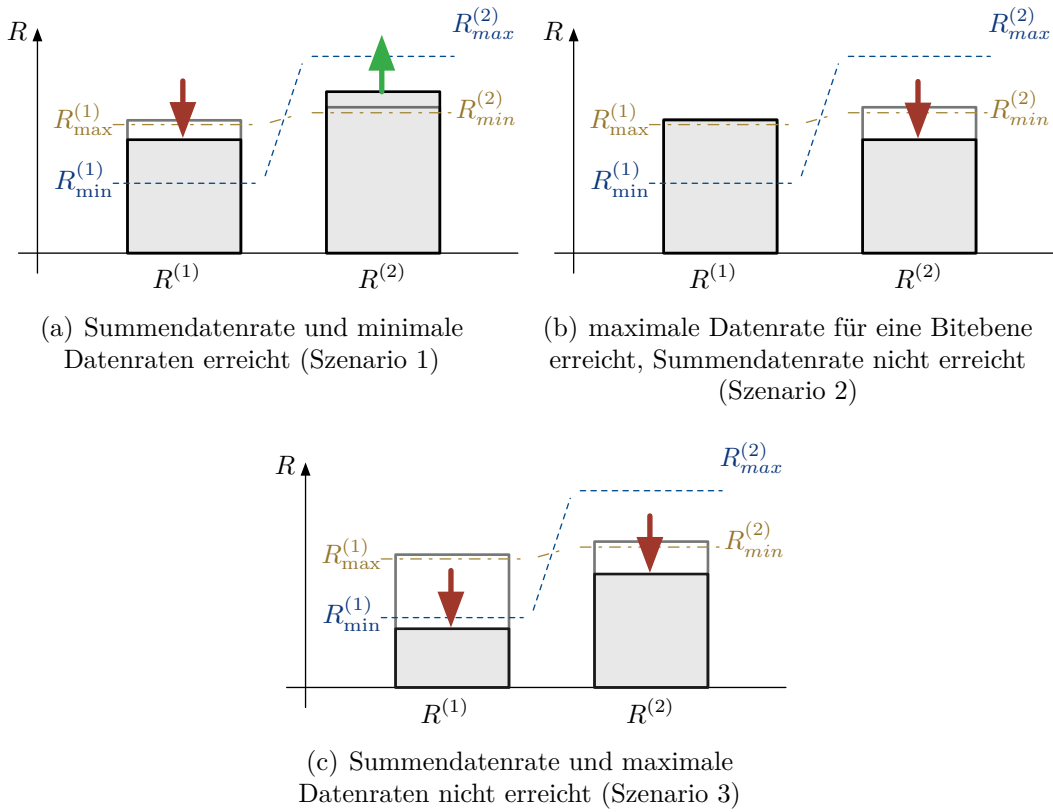


Abbildung 7.5.: Szenarios für die flexible Decodierung, unterschiedliche Verteilung der Datenrate im Vergleich zur Bitebenen-weisen Decodierung (Abb. 7.4), zwei Bitebenen ($M = 2$)

7.4. Bitebenen übergreifende Decodierung

7.4.1. Lösungsansatz

Die Bitebenen übergreifende Decodierung stellt einen Lösungsansatz für das Szenario 1 dar, in welchem die Gesamtdatenrate für die erfolgreiche Decodierung ausreichend, diese jedoch ungünstig zwischen den Bitebenen verteilt ist. Die Decodierung der Bitebenen nacheinander ist in diesem Fall nicht möglich, da mit der Decodierung keiner der Bitebenen gestartet werden kann. Zur Lösung des Problems wird hier eine Bitebenen übergreifende Decodierung vorgeschlagen.

Die schrittweise Decodierung der Bitebenen, wie sie in *ρDeViCe* verwendet wird (Abb. 5.8, S. 111), bearbeitet eine Bitebene nach der anderen. Tritt ein Decodierungsversagen auf, so können die weiteren Bitebenen meist nicht decodiert werden.

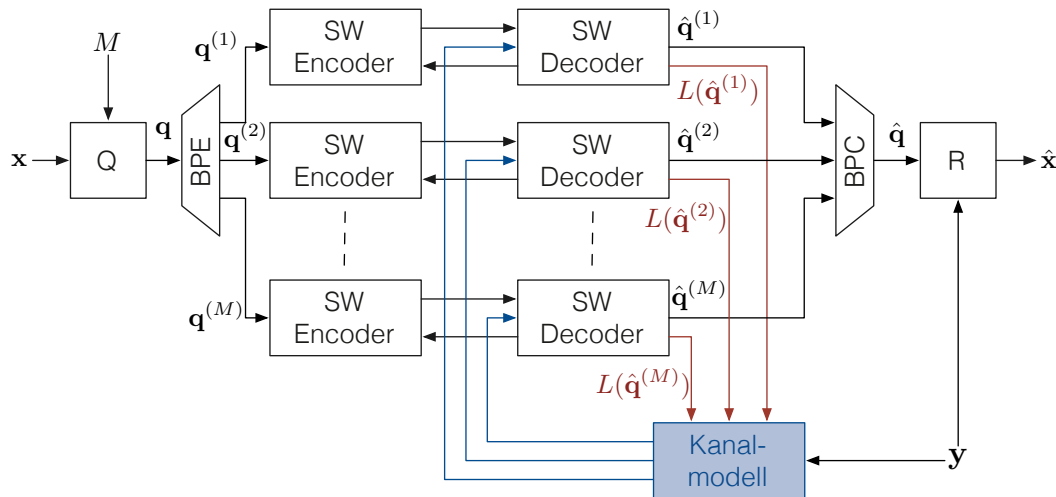


Abbildung 7.6.: Bitebenen übergreifende Decodierung durch Austausch von Zuverlässigkeitsinformationen

Die Bitebenen übergreifende Decodierung [Hänsel u. Müller 2009c] löst das Problem, indem die Bitebenen parallel decodiert werden. Um dies zu ermöglichen, werden nicht nur die hart entschiedenen decodierten Bitebenen $\hat{\mathbf{q}}^{(b)}$ der Quantisierungssymbole ausgetauscht, sondern auch die Zuverlässigkeitsinformation in Form der posteriori LLR-Werte $L(\hat{\mathbf{q}}^{(b)})$ (Abb. 7.6, Gl. 7.1). Dieser Prozess wird iterativ

wiederholt, bis eine erfolgreiche Decodierung möglich ist oder eine maximale Anzahl Wiederholungen erreicht ist.

Dieser Ansatz entspricht der bekannten Super-Turbo Decodierung [Garcia-Frias u. Zhao 2001][Zhao u. Garcia-Frias 2002], wobei die statistischen Abhängigkeiten zwischen den Bitebenen ausgenutzt werden.

Später wurde in [Luong u. a. 2011] die Nutzung der Bitebenen übergreifenden Decodierung für LDPC Codes zur Verbesserung der Leistungsfähigkeit in Systemen mit voller Rückkanalnutzung untersucht.

$$L_c(\hat{q}_i^{(b)}) = \ln \frac{p'_{Y|Q}(y_i|q_i^{(b)} = 0)}{p'_{Y|Q}(y_i|q_i^{(b)} = 1)} \quad (7.1)$$

$$p'_{Y|Q}(y_i|q_i^{(b)} = 0) = \sum_{q|q^{(b)}=0} \underbrace{\left\{ \prod_{k \neq b} \Pr(q^{(k)} = \hat{q}_i^{(k)}) \right\}}_{\Pr(q=\hat{q}_i|q^{(b)}=0)} \cdot p_{Y|Q}(y_i|q) \quad (7.2)$$

$$\Pr(\hat{q}_i^{(b)} = 0) = \frac{e^{L(\hat{q}_i^{(b)})}}{1 + e^{L(\hat{q}_i^{(b)})}} = 1 - \Pr(\hat{q}_i^{(b)} = 1) \quad (7.3)$$

Der wesentliche Unterschied zur schrittweisen Decodierung der Bitebenen besteht somit darin, dass bei Anwendung des Modells die Quantisierungsintervalle nicht aufgrund der bekannten Bitebenen $\hat{q}_i^{(b)}$ ausgeblendet werden. Vielmehr werden die bekannten Wahrscheinlichkeiten $\Pr(q = \hat{q}_i|q^{(b)} = 0)$ bzw. $\Pr(q = \hat{q}_i|q^{(b)} = 1)$ verwendet, um die Quantisierungsintervalle entsprechend zu wichten (Gl. 7.2, Abb. 7.7).

Das Beispiel in Abbildung 7.7 zeigt die Bitebenen übergreifende Decodierung, beginnend mit der Bitebene 1. Im ersten Schritt kann keine Wichtung oder Ausblendung von Quantisierungsintervallen vorgenommen werden, da keine posteriori LLR-Werte bzw. decodierte Bitebenen vorliegen. Im Beispiel hat die Decodierung der ersten Bitebene eine 0 mit einer Wahrscheinlichkeit von 0.8 ergeben, womit die Quantisierungssymbole $q_i = 0$, $q_i = 1$ mit 0.8 und die weiteren Symbole mit 0.2 gewichtet werden. Dieser Prozess wird iterativ für alle weiteren Bitebenen fortgesetzt und bei erfolgreicher Decodierung oder bei Erreichen einer maximalen Iterationsanzahl $\nu > \nu_{max}$ gestoppt.

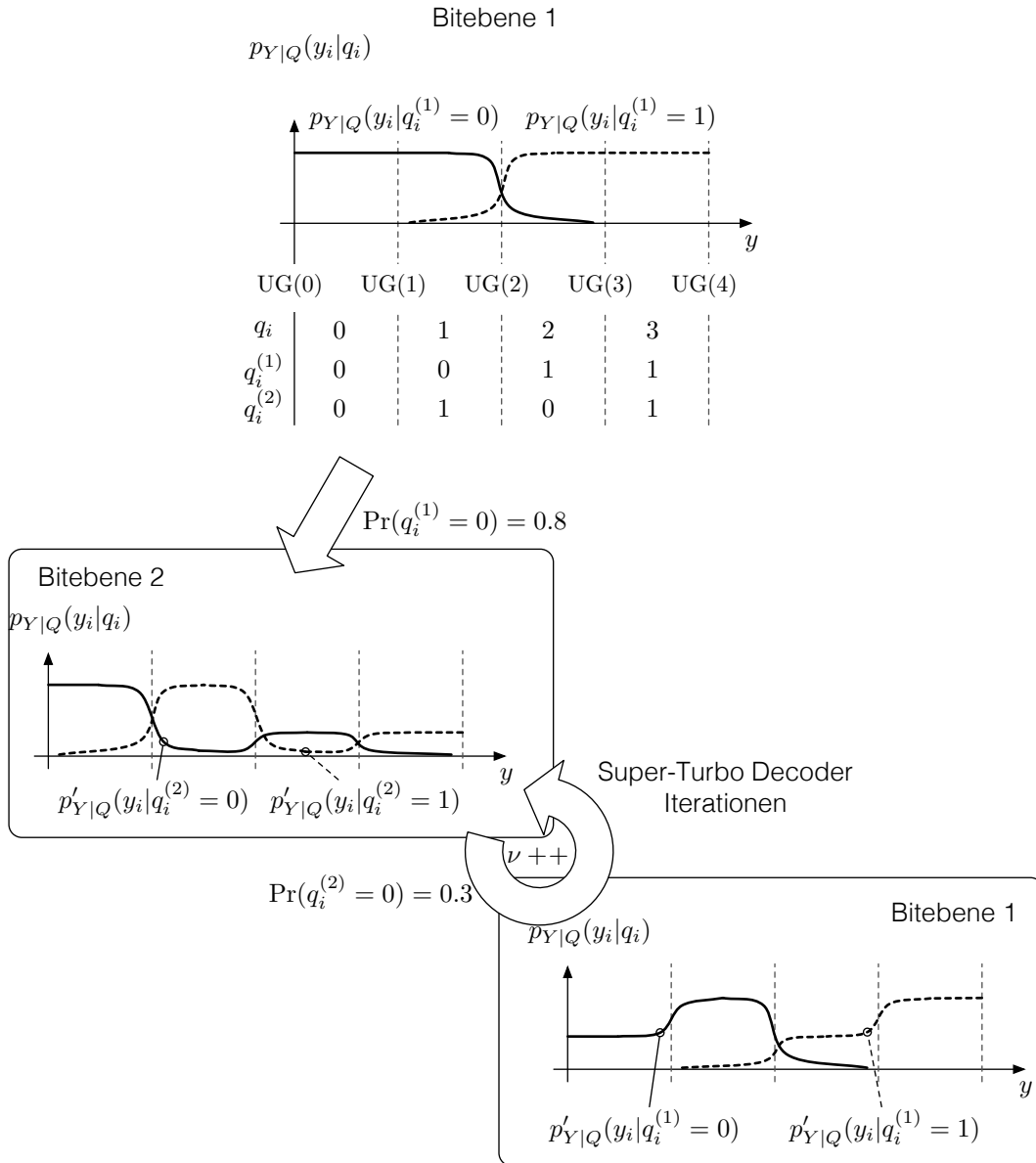


Abbildung 7.7.: Iterative Bitebenen übergreifende Decodierung, beginnend mit Bitebene 1 (Super-Turbo Decoder)

Mit der Bitebenen übergreifenden Decodierung wird ein Informationsaustausch zwischen den Decodern im gewissen Rahmen ermöglicht. Für jede Bitebene muss eine Datenrate $R^{(b)}$ übertragen werden, welche dem Informationsgehalt dieser entspricht, wenn alle anderen Bitebenen bekannt sind $\forall b, R^{(b)} \geq R_{min}^{(b)}$ (Gl. 7.4). Weiterhin muss die Summendatenrate, welche zur erfolgreichen Codierung notwendig ist, erreicht

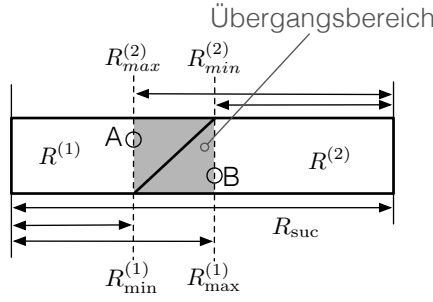


Abbildung 7.8.: Verteilung der Datenrate zwischen den Bitebenen bei gemeinsamer Decodierung, zwei Bitebenen $M = 2$, Decodierreihenfolge: Punkt A $2 \rightarrow 1$, Punkt B $1 \rightarrow 2$

worden sein, $\sum_k R^{(k)} \geq R_{suc}$.

$$R_{min}^{(b)} \geq H(Q^{(b)}|Y, Q^{(1)}, \dots, Q^{(b-1)}, Q^{(b+1)}, \dots, Q^{(M)}) \quad (7.4)$$

$$R_{max}^{(b)} \geq H(Q^{(b)}|Y) \quad (7.5)$$

Damit lassen sich die in Abschnitt 7.3 aufgestellten Bedingungen für die erfolgreiche Decodierung aufweichen. Neben den in Abbildung 7.8 dargestellten Ratenkombinationen A,B lassen sich durch die übergreifende Decodierung alle Kombinationen innerhalb des grauen Übergangsbereiches erreichen. Die Ratenkombinationen A,B entstehen durch Variation der Decodierreihenfolge in der schrittweisen Decodierung, dabei wird die Datenrate geringer, wenn die Bitebene später decodiert wird, da dann mehr decodierte Bitebenen vorliegen.

7.4.2. Simulationsergebnisse

Durch die Bitebenen übergreifende Decodierung wird der Decoder in die Lage versetzt, eine Decodierung durchzuführen, auch wenn für keine der Bitebenen ausreichend Korrektursymbole übertragen wurden, um diese als Erstes zu decodieren.

Abbildung 7.9 zeigt die Ratenbereiche, welche durch die Bitebenen-weise Decodierung erreicht werden können (grau). Die Kombinationen $(R_{1,min}, R_{2,max})$ bzw. $(R_{1,max}, R_{2,min})$ werden üblicherweise verwendet, da so die geringste Gesamtdatenrate R_{suc} erreicht wird. Die Bitebenen übergreifende Decodierung ermöglicht es,

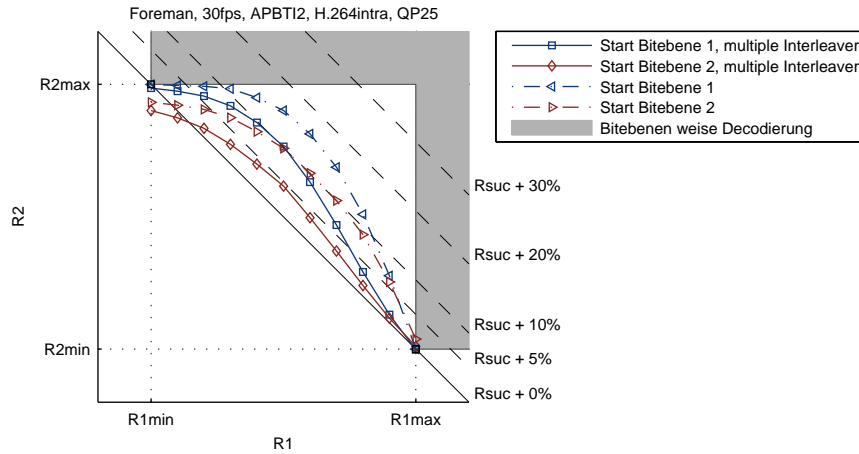


Abbildung 7.9.: Zur erfolgreichen Decodierung notwendige Datenraten R_1 , R_2 bei gemeinsamer Decodierung der Bitebenen, **Foreman**, QCIF, 30 fps, APBTI2, Key-Frames: H.264intra, QP25, $M = 2$

bei weiteren Kombinationen (R_1, R_2) erfolgreich zu decodieren, wobei nur eine geringe Erhöhung der Gesamtdatenrate von 5% eintritt. Diese höhere Flexibilität kann durch eine Encoder-seitige Ratenkontrolle genutzt werden, da so das Verhältniss zwischen den Datenraten der einzelnen Bitebenen nicht mehr exakt bestimmt werden muss.

Weiterhin kann gezeigt werden, dass die übergreifende Decodierung eine höhere Leistungsfähigkeit zeigt, wenn die Super-Turbo Decodierung mit der zweiten also geringwertigeren Bitebene gestartet wird. Darüber hinaus wurde untersucht, inwieweit die Dekorrelation zwischen den Bitebenen durch zusätzliche Interleaver notwendig ist. Die Simulationsergebnisse zeigen, dass durch die Verwendung multipler Interleaver die Leitungsfähigkeit weiter gesteigert werden kann (Abb.: 7.9).

Eine Betrachtung für drei und mehr Bitebenen wurde hier nicht durchgeführt. Die Bitebenen übergreifende Decodierung lässt sich hier jedoch auch anwenden, jedoch muss die Reihenfolge der Bearbeitung gesondert untersucht werden.

7.4.3. Fazit

In diesem Abschnitt wurde die Bitebenen übergreifende Decodierung vorgeschlagen, um eine flexible Verteilung der Datenrate zwischen den Bitebenen zu erreichen. Damit kann eine erfolgreiche Decodierung für das Szenario 1 durchgeführt werden

(Abb. 7.5(a), S. 150). Es konnte gezeigt werden, dass für zwei Bitebenen ($M = 2$) die Decodierung auch dann möglich ist, wenn für keine der Bitebenen die maximale Datenrate übertragen wurde. Dabei ergab sich bei Verwendung von multiplen Interleavern, zur Dekorrelation der Bitebenen, eine Erhöhung der Gesamtdatenrate von 5 %.

Dieser Ansatz vereinfacht die Encoder-seitige Ratenkontrolle dahin gehend, dass nicht für jede Bitebene die exakte Datenrate bestimmt werden muss. Damit sind die Nachteile gegenüber der symbolweisen Codierung, bei der keine Zerlegung in Bitebenen durchgeführt wird, aufgehoben. Weiterhin bleibt die Möglichkeit, die Decodierung bei einer bestimmten Bitebene abubrechen, erhalten.

7.5. Soft-Rekonstruktion

Mit der im vorangegangenen Abschnitt 7.4 betrachteten Bitebenen übergreifenden Decodierung wurde der Austausch von Zuverlässigkeitsinformationen eingeführt. Dies soll nun auf die Rekonstruktion erweitert werden.

Die Soft-Rekonstruktion [Hänsel u. Müller 2009b] verwendet neben den erfolgreich decodierten Bitebenen auch die Zuverlässigkeitsinformation $L(q^{(b)})$ von Bitebenen, welche nicht erfolgreich decodiert werden konnten. Die Idee besteht darin, die Tendenz der Log-Likelihood Werte hin zum richtigen Quantisierungssymbol zu nutzen, um eine Verbesserung der Rekonstruktionsqualität zu erhalten.

Der gleiche Ansatz wurde später in [Brites u. Pereira 2011] auf die Rekonstruktion von DCT-Koeffizienten in einem DVC System mit Encoder-seitiger Ratenkontrolle angewendet.

Die Soft-Rekonstruktion wurde für das Szenario 2 und 3 (S. 148) entworfen, da hier einige oder alle Bitebenen aufgrund zu geringer Datenrate nicht decodiert werden können. In diesem Fall tritt eine schlechte RD-Performanz ein, da eine hohe Datenrate übertragen wurde, diese jedoch nicht zur Verbesserung des Rekonstruktionsergebnisses führt. In diesem Fall kann der Einsatz der Soft-Rekonstruktion zu einer leichten Verbesserung des Rekonstruktionsergebnisses führen.

Die Umsetzung der Soft-Rekonstruktion erfolgt in Anlehnung an die in *qDeViCe* verwendete MMSE-Rekonstruktion (Gl. 5.10, S. 115). Der wesentliche Unterschied

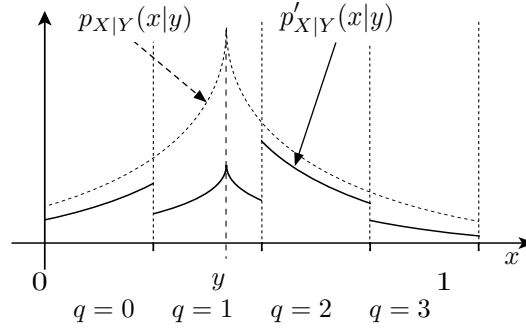


Abbildung 7.10.: Wichtung der Quantisierungsintervalle bei der Soft-Rekonstruktion

besteht nun darin, dass nach der Decodierung kein einzelnes Quantisierungsintervall ausgewählt werden kann, sondern die Intervalle entsprechend ihrer Wahrscheinlichkeit gewichtet werden (Gl. 7.8, 7.6, Abb. 7.10). Die Wahrscheinlichkeit der Quantisierungsintervalle ergibt sich aus den Wahrscheinlichkeiten der einzelnen Bitebenen (Gl. 7.7).

$$\begin{aligned}\hat{x}_i &= E\{x_i | \Pr(\hat{q}_i), y_i\} \\ &= \frac{\int_{-\infty}^{+\infty} x \cdot p'_{X|Y}(x|y_i) dx}{\int_{-\infty}^{+\infty} p'_{X|Y}(x|y_i) dx}\end{aligned}\quad (7.6)$$

$$\Pr(\hat{q}') = \prod_b \Pr(\hat{q}^{(b)} = \hat{q}'^{(b)}) \quad (7.7)$$

$$p'_{X|Y}(x|y) = \Pr(\hat{q} = Q(x)) \cdot p_{X|Y}(x|y) \quad (7.8)$$

Die vorläufigen Simulationsergebnisse haben gezeigt, dass die Soft-Rekonstruktion zu einer leichten Verbesserung des Rekonstruktionsergebnisses führt, wenn nur die geringwertigste Bitebene (LSB) nicht decodiert werden kann. Tritt jedoch der Fall ein, dass auch höherwertige Bitebenen oder alle Bitebenen nicht decodiert werden können, so führt die Soft-Rekonstruktion zu keiner Verbesserung der Leistungsfähigkeit in einem pixelbasierten DVC Codec.

Ursache hierfür können die verwendeten Zuverlässigkeitsinformationen darstellen. Die Bestimmung der Wahrscheinlichkeit eines Symbols aus dem posteriori LLR-

Wert ist nicht immer zuverlässig. Hier ist die Betrachtung weiterführende Ansätze notwendig. In [Hänsel u. Müller 2010] wurde die Extraktion von Zuverlässigkeitsinformation auf Basis modifizierter Abbruchkriterien für Turbocodes [Skorupa u. a. 2009] untersucht. Diese Ansätze sind bisher nicht in die Soft-Rekonstruktion oder andere Verfahren der flexiblen Decodierung eingeflossen.

7.6. Fehlerlokalisierung und -verschleierung

Die Szenarien 2 und insbesondere 3 (S. 148) stellen eine große Herausforderung dar. In diesen Fällen können einzelne oder alle Bitebenen nicht decodiert werden, damit ist keine herkömmliche Rekonstruktion möglich. Hier soll nun ein Ansatz vorgestellt werden, welcher die Trennung zwischen verlustloser und verlustbehafteter Verteilter Codierung aufbricht.

Die Kanalcodierung hat im Wesentlichen zwei Aufgaben. Zum einen dienen Kanalcodes dem Fehlerschutz (z.B. Faltungscodes), zum anderen können Kanalcodes auch zur Fehlererkennung eingesetzt werden (z.B. CRC), um festzustellen, ob ein oder mehrere Fehler in einer Symbolfolge enthalten sind. Die Fehlererkennung setzt einen wesentlichen schwächeren Kanalcode als die Fehlerkorrektur voraus, womit die Coderate R_c höher und die Anzahl der übertragenen Korrektursymbole $\mathbf{c}_{p1}, \mathbf{c}_{p2}$ geringer ausfallen kann.

Neben diesen beiden häufigen Einsatzfeldern werden Kanalcodes auch zur Lokalisierung von Fehlern eingesetzt (ELC, *error locating codes*). Je nachdem, wie leistungsfähig das verwendete Codierungsverfahren ist, kann eine feingranuläre oder unscharfe Fehlerlokalisierung durchgeführt werden. Wenn es nun möglich ist, unter Verwendung der übertragenen Korrektursymbole $\mathbf{c}_{p1}, \mathbf{c}_{p2}$ die Fehlerposition in der Seiteninformation \mathbf{Y}_i zu bestimmen, so kann an diesen Orten gezielt eine Fehlerverschleierung eingesetzt werden, um die durch den Betrachter wahrgenommene Qualität zu verbessern.

Der vorgeschlagene WZ-Decoder (Abb. 7.11) beinhaltet neben den klassischen Komponenten zur WZ-Decodierung zusätzlich ein Modul zur Lokalisierung der Fehlerpositionen und Verfahren zur Verschleierung bzw. Überzeichnung von Fehlern.

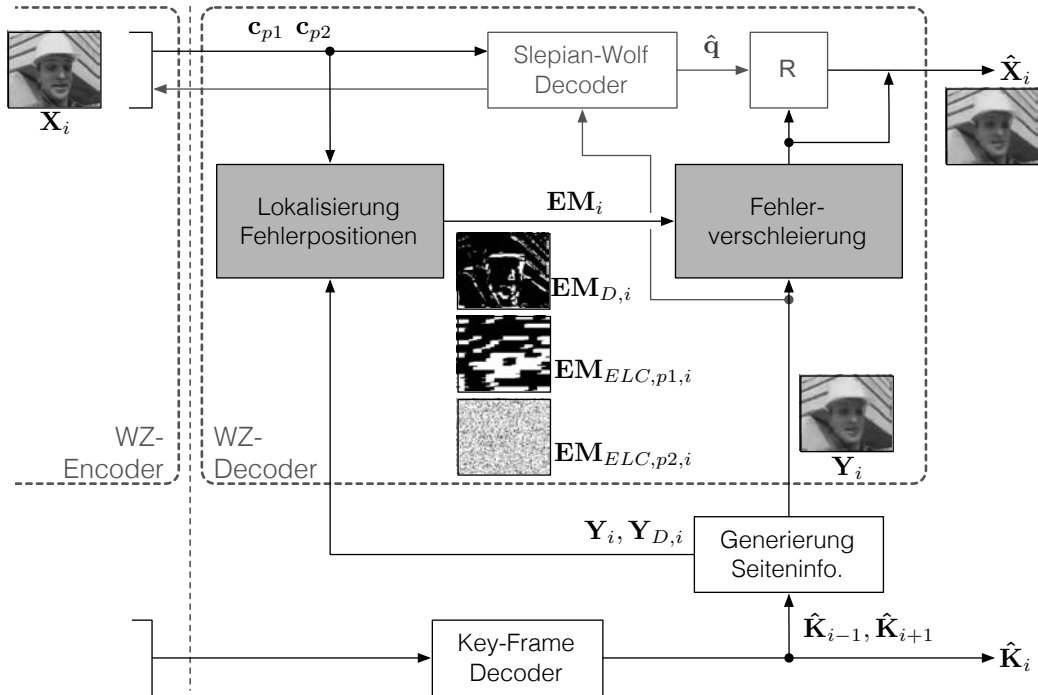


Abbildung 7.11.: Flexibler Wyner-Ziv Decoder mit Fehlerlokalisierung und Fehlerverschleierung

Fehlerlokalisierung mittels Verfahren der Verteilten Quellencodierung wurde in [Lin u. a. 2007, 2008] für das Watermarking-anwendungsszenario betrachtet. Weiterhin wurden in [Said 1995; Kitakami u. Sano 2005] Verfahren zur Lokalisierung von Blockfehlern vorgestellt.

Der Einsatz einer Fehlerverschleierung zur Minderung der Auswirkungen von Übertragungsfehlern ist für die konventionelle [Seiler u. Kaup 2008] und Verteilte Videocodierung [Ye u. a. 2008] bekannt. Dabei unterscheidet man in zeitliche und örtliche Fehlerverschleierung. Die Erhöhung der Leistungsfähigkeit einer DVC Codec kann auch durch Methoden der Fehlerverschleierung erreicht werden [Sofke u. a. 2009]. Zur örtlichen Fehlerverschleierung kann *Image Inpainting* [Tschumperlé 2002; Bertalmio u. a. 2000] angewandt werden, wobei das PSNR zur späteren Leistungsbewertung ungeeignet ist.

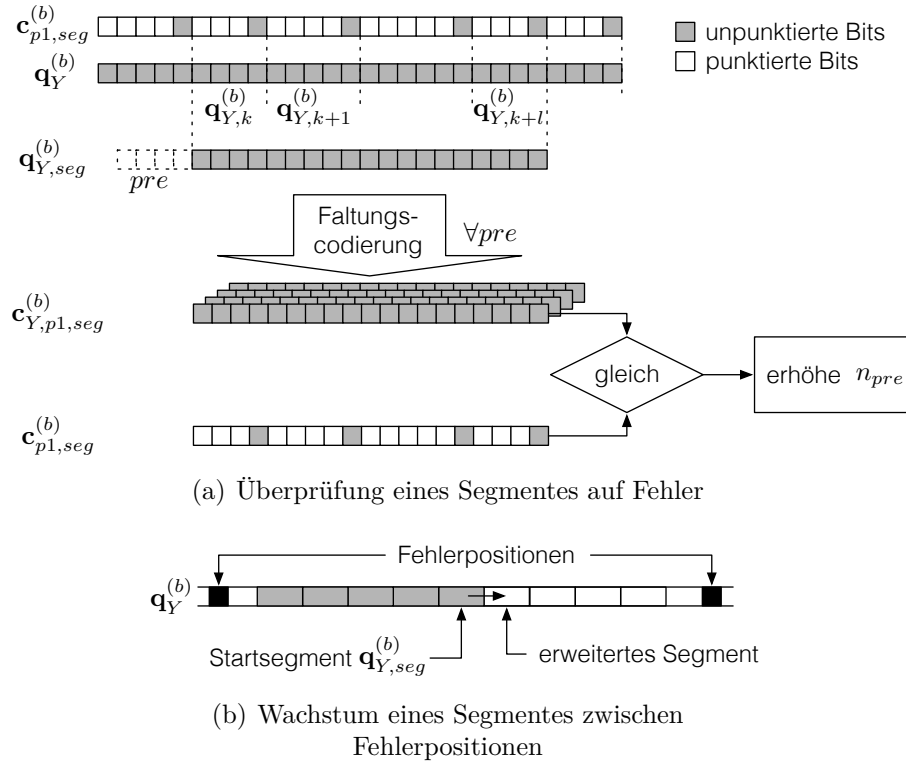


Abbildung 7.12.: Bestimmung von potentiellen Fehlersegmenten basierend auf den empfangenen Paritätssymbolen c_{p1}, c_{p2}

7.6.1. Unscharfe Bestimmung der Fehlerposition

Zur Bestimmung von Fehlerpositionen in der Seiteninformation \mathbf{Y}_i werden zum einen die übertragenen Korrektursymbole c_{p1}, c_{p2} und zum anderen die Differenz $\mathbf{Y}_{D,i}$ zwischen den bewegungskompensierten Key-Frames verwendet (Gl. 7.10) [Hänsel u. Müller 2009a].

Die beiden in *ρDeViCe* verwendeten parallel verketteten Faltungscodes werden einzeln zur Lokalisierung von Fehlern verwendet. Dazu wird die Seiteninformation quantisiert und in Bitebenen zerlegt $q_Y^{(b)}$. Zur Fehlerlokalisierung werden Mikrosegmente $q_{Y,k}^{(b)}$ entsprechend des Punktierungsschemas gebildet, sodass jedes Mikrosegment genau ein unpunktiertes Korrektursymbol $c_{p1,k}^{(b)}$ bzw. $c_{p2,k}^{(b)}$ enthält (Abb. 7.12(a)).

Einzelne Mikrosegmente können zu einem Segment $q_{Y,seg}^{(b)}$ zusammengefasst werden, für welches mittels Faltungscodierung die Paritätssymbole $c_{Y,p1}^{(b)}$ bestimmt und mit den empfangenen Symbolen verglichen werden. Sind diese gleich, so trat in diesem

Segment wahrscheinlich kein Fehler auf. Da der Zustand des Faltungscodierers zu Beginn des Segmentes unbekannt ist, wird eine Präambel pre entsprechend des Gedächtnisses des Faltungscodierers angefügt. Ein Segment gilt nur dann als fehlerfrei, wenn für mindestens eine Präambel die am Encoder erzeugten $\mathbf{c}_{p1,seg}^{(b)}$ mit den am Decoder aus der Seiteninformation erzeugten Paritätssymbolen $\mathbf{c}_{Y,p1,seg}^{(b)}$ übereinstimmen ($n_{pre} \geq 1$). Wurde ein fehlerfreies Segment gefunden, so werden sukzessiv weitere Mikrosegmente angefügt (Abb. 7.12(b)), um möglichst lange Segmente mit vielen unpunktierten Bits zu erhalten. Ein Bildpunkt wird nur dann als fehlerhaft angesehen, wenn dieser aus Sicht beider Faltungscoder einem fehlerhaften Segment zugeordnet wurde (Gl. 7.9).

$$\mathbf{EM}_{ELC,i}(x, y) = \mathbf{EM}_{ELC,p1,i}(x, y) \wedge \mathbf{EM}_{ELC,p2,i}(x, y) \quad (7.9)$$

Die Granularität der Fehlerlokalisierung dieses Ansatzes hängt direkt von der übertragenen Datenrate ab. Werden mehr Daten übertragen, so bilden sich kleinere Mikrosegmente, womit die Fehlerpositionen genauer lokalisiert werden können.

Die Fehlerlokalisierung kann weiter verbessert werden, indem die Differenz $\mathbf{Y}_{D,i}$ der bewegungskompensierten Key-Frames $\mathbf{K}_{i-1}^{-\mathbf{MVF}}$ bzw. $\mathbf{K}_{i+1}^{\mathbf{MVF}}$ mit einbezogen wird (Gl. 7.11). Diese stellt ein Maß für die Zuverlässigkeit der Bewegungsschätzung dar. Hohe Differenzen zwischen den Key-Frames werden in der entsprechenden Fehlerkarte $\mathbf{EM}_{D,i}$ als fehlerhafte Position vermerkt.

$$\mathbf{Y}_{D,i}(x, y) = \left\{ \mathbf{K}_{i-1}^{-\mathbf{MVF}}(x, y) - \mathbf{K}_{i+1}^{\mathbf{MVF}}(x, y) \right\} \quad (7.10)$$

$$\mathbf{EM}_{D,i}(x, y) = \begin{cases} 1 & : \mathbf{Y}_{D,i}(x, y) > th \\ 0 & : sonst \end{cases} \quad (7.11)$$

$$\mathbf{EM}_i(x, y) = \mathbf{EM}_{D,i}(x, y) \wedge \mathbf{EM}_{ELC,i}(x, y) \quad (7.12)$$

7.6.2. Verschleierung von Fehlern

Die im vorangegangenen Abschnitt bestimmten Fehlerpositionen können ausgeblendet werden und mittels örtlicher Fehlerverschleierung, wie zum Beispiel *Image In-*

painting [Tschumperlé 2002][Bertalmio u. a. 2000], gefüllt werden.

Simulationen haben gezeigt, dass so die Leistungsfähigkeit bisher nur im geringen Maße verbessert werden kann.

7.6.3. Fazit

Der vorgeschlagene Ansatz, die für die SW-Codierung verwendeten Kanalcodes auch für die Fehlerlokalisierung und späterer Fehlerverschleierung einzusetzen, kann auch im Fall eines Decoderversagens angewendet werden. Vorläufige Simulationen haben gezeigt, dass damit die Rekonstruktionsqualität objektiv (PSNR) und subjektiv geringfügig verbessert werden kann.

Ein wesentliches Problem stellt jedoch der Kompromiss zwischen hoher Leistungsfähigkeit des parallel verketteten Faltungscodes für die Fehlerkorrektur und der Fehlerlokalisierung dar. Eine leistungsfähige Fehlerlokalisierung wird erreicht, wenn Bündelfehler auftreten, dies mindert hingegen die Leistungsfähigkeit der Fehlerkorrektur. Die Bündelfehler können gezielt durch den Einsatz eines Interleavers zerstreut werden, womit der Auswahl des Interleavers eine zentrale Bedeutung zukommt.

Aus diesem Grunde ist es notwendig, leistungsfähige Codierungsverfahren zu finden, welche eine hohe Leistungsfähigkeit sowohl bezüglich der Fehlerkorrektur als auch Fehlerlokalisierung zeigen. Weiterhin ist es auch denkbar, zur Fehlerlokalisierung auf weitere Merkmale im Verlauf des Decodierungsprozesses zurückzugreifen [Hänsel u. Müller 2010].

7.7. Zusammenfassung

Dem Rückkanal kommt in der Verteilten Videocodierung eine zentrale Bedeutung zu. Da dieser in vielen Anwendungsszenarien nicht zur Verfügung steht, kann in diesen die Verteilte Videocodierung nur bedingt eingesetzt werden. Zur Lösung des Problems gibt es in der Literatur im Wesentlichen zwei Ansätze, die Encoder- und Decoder-seitige Ratenkontrolle, wobei die Nutzung des Rückkanals verringert oder ganz vermieden wird. Darüber hinaus wurden in diesem Kapitel Ansätze zur

flexiblen Decodierung vorgeschlagen, welche dem Decoder mehr Möglichkeiten im Umgang mit den empfangenen Daten bieten.

Die entwickelte Bitebenen übergreifende Decodierung ermöglicht einen flexiblen Informationsaustausch zwischen den zu decodierenden Bitebenen, womit, auch bei ungünstiger Verteilung der Datenrate zwischen den Bitebenen, eine Decodierung möglich ist. Weiterhin wurde für den Fall, dass aufgrund zu geringer Gesamtdatenrate keine Decodierung möglich ist, die Soft-Rekonstruktion und die Fehlerverschleierung, basierend auf unscharfer Fehlerlokalisierung, vorgeschlagen.

Die hier entwickelten Lösungen stellen Ansätze für die Vermeidung des Rückkanals dar. Die in der Literatur sowie in *qDeViCe* verwendete Systemstruktur ist nach aktuellem Kenntnisstand nicht für die rückkanalfreie Decodierung geeignet und bedarf einer Umstrukturierung, wie es im Ansatz zur Fehlerlokalisierung und Fehlerverschleierung (Abschn. 7.6) beispielhaft gezeigt wurde.



Zusammenfassung und Ausblick

8.1. Zusammenfassung

Die Zielsetzung dieser Arbeit bestand in der Entwicklung eines Videocodierungssystems mit geringer Encoderkomplexität nach den Prinzipien der Verteilten Codierung. Die geringe Komplexität des Encoders, die Steigerung der Leistungsfähigkeit und die Flexibilität der Decodierung standen im Mittelpunkt der Entwicklungen und Untersuchungen.

Die Verwendung einer örtlichen Transformation, wie zum Beispiel der diskreten Cosinus-Transformation (DCT), erhöht die Komplexität des Codierungssystems im Vergleich zu einem pixelbasierten Codec. Weiterhin kann die Komplexität weiterer Komponenten des Encoders durch die Anwendung eines transformationsbasierten Ansatzes reduziert werden. Die sehr geringe Encoderkomplexität und hohe Flexibilität der pixelbasierten Systeme konnte jedoch nicht erreicht werden. Basierend auf diesen Erkenntnissen wurde konsequent der pixelbasierte Ansatz verfolgt.

Die Qualität der Seiteninformation hat starken Einfluss auf die Leistungsfähigkeit des Gesamtsystems. Basierend auf bekannten Verfahren zur temporalen Interpolation wurden neue Verfahren für Herausforderungen, wie sehr schnelle oder inhomogene Bewegungen, entwickelt.

Es hat sich gezeigt, dass durch einen zweistufigen Prozess aus Abschätzung der globalen Bewegung und verfeinerter pixelbasierter Bewegungsschätzung, eine wesentliche Verbesserung der Seiteninformation, für Sequenzen mit schnellen Bewegungen, erreicht werden kann. Zur robusten Abschätzung der globalen Bewegung wurde auf die Bewegung markanter Bildbereiche zurückgegriffen.

Dieser Ansatz der robusten Bewegungsabschätzung wurde auch auf Sequenzen mit einer inhomogenen Bewegungsstruktur übertragen. Hier konnte durch multiple Bewegungshypothesen eine Verbesserung der Bewegungsschätzung erreicht werden, ohne den Aufwand für die Bewegungsschätzung zu erhöhen. Die geschickte Nutzung markanter Bildbereiche führte auch zu einer Verringerung der Komplexität der temporalen Interpolation.

Weiterhin konnte die Qualität der Seiteninformation, in Sequenzen mit starken Auf- und Verdeckungsbereichen, durch eine adaptive zeitliche Inter- und Extrapolation sowie durch eine örtliche Extrapolation des Bewegungsvektorfeldes verbessert werden.

Die zweite wichtige Komponente eines Verteilten Videocodierungssystems, die Wyner-Ziv Codierung, wurde mittels der leistungsfähigen Turbo-Codierung realisiert. Es konnte gezeigt werden, dass die Turbo-Codierung für lange Symbolfolgen in den für die Verteilte Videocodierung relevanten Bereichen eine bessere Leistungsfähigkeit als die LDPCA-Codierung erreicht. Weiterhin muss die Berücksichtigung von ungleichmäßig verteilten Eingangssymbolen (z.B. Grauwerte im Eingangsbild) im Rahmen der Slepian-Wolf Codierung erfolgen, um eine Steigerung der Leistungsfähigkeit zu erreichen.

Die Frames der Videosequenzen wurden abwechselnd als Key- und Wyner-Ziv Frames codiert, wobei die RD-Performanz der Key-Frames die Leistungsfähigkeit des Gesamtsystems beeinflusst. Es konnte gezeigt werden, dass die Seiteninformationsqualität nicht von der Art der Codierung der Key-Frames, als viel mehr von deren Qualität und der Methode zur Generierung der Seiteninformation, abhängt.

Das entwickelte Gesamtsystem *oDeViCe* erreichte, für Sequenzen mit geringer und moderater Bewegungskomplexität, eine mit H.264intra und HEVCintra vergleichbare Leistungsfähigkeit. Für Sequenzen mit komplexen Bewegungsstrukturen wurde zwar eine wesentliche Steigerung der Leistungsfähigkeit erreicht, die Performanz der konventionellen Verfahren konnte jedoch nicht erreicht.

Die flexible Decodierung wurde zur Reduktion bzw. Vermeidung der Nutzung des Rückkanals vorgeschlagen. Es konnte durch eine Bitebenen übergreifende Decodierung eine Umverteilung von Information erreicht werden, wobei die Gesamtdatenrate um maximal 5% erhöht wurde. Weitere Vorschläge, wie die Soft-Rekonstruktion und die unscharfe Fehlerpositionierung mit nachfolgender Fehlerverschleierung, stellen Ansätze für eine weiterführende flexible Decodierung dar.

8.2. Ausblick

Weitere Ansätze für die Verbesserung der Leistungsfähigkeit beziehen sich auf die Adaption der Codierungsparameter. So könnten die Quantisierungsparameter der Key- und Wyner-Ziv Frames dynamisch aufeinander abgestimmt werden, um eine höhere RD-Performanz sowie eine höhere subjektive Qualität zu erreichen.

Eine große Herausforderung stellt nach wie vor die Vermeidung des Rückkanals dar. Besonders auf diesem Gebiet gibt es großen Forschungsbedarf. Der entwickelte Ansatz der flexiblen Decodierung sollte in dem Sinne fortgesetzt werden, sodass neben der Turbo-Decodierung parallel verketteter Faltungscodes weitere Rekonstruktionsverfahren gefunden werden, welche die Rekonstruktionsqualität verbessern. Diese Entwicklungsrichtung sollte nicht auf die Turbo-Codierung beschränkt bleiben.

Ein weiterführender Ansatz sollte die Erweiterung des Codierungssystems, in das Forschungsgebiet der Interaktiven Codierung hinein, berücksichtigen, wobei die intensive Nutzung des Rückkanals dem Anwendungsszenario entspricht. Es bleibt zu untersuchen, wie der Rückkanal, neben der Ratenkontrolle, zur Verbesserung der Leistungsfähigkeit eingesetzt werden kann.

Die Verteilte Videocodierung kann hinsichtlich der RD-Performanz nur schwer mit den konventionellen Videocodierungsverfahren konkurrieren. Besonderes Forschungsinteresse sollten Systeme erhalten, welche eine Kombination konventioneller und Verteilter Videocodierung anstreben, um so die Vorteile beider Ansätze zu nutzen. In [Sullivan u. a. 2006] wurde, in Bezug auf die Zukunft der Videocodierung, die Übernahme von Verfahren und Methoden aus der Verteilten Videocodierung in konventionelle Videocodierungssysteme betont.

Literaturverzeichnis

Aaron u. Girod 2002

AARON, Anne ; GIROD, Bernd: Compression with Side Information Using Turbo Codes. In: *Proc. IEEE Data Compression Conference (DCC)*. Snowbird, Utah, USA, 2002

Aaron u. a. 2002

AARON, Anne ; ZHANG, Rui ; GIROD, Bernd: Wyner-Ziv Coding of Motion Video. In: *Proc. Asilomar Conference on Signals, Systems, and Computers*. Pacific Grove, CA, USA, 2002

Aaron u. a. 2003a

AARON, Anne ; SETTON, Eric ; GIROD, Bernd: Towards practical Wyner-Ziv coding of video. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Barcelona, Spain, 2003

Aaron u. a. 2003b

AARON, Anne ; RANE, Shantanu ; ZHANG, Rui ; GIROD, Bernd: Wyner-Ziv Coding for Video: Applications to Compression and Error Resilience. In: *Proc. IEEE Data Compression Conference (DCC)*. Snowbird, Utah, USA, 2003

Aaron u. a. 2004a

AARON, Anne ; RANE, Shantanu ; SETTON, Eric ; GIROD, Bernd: Transform-domain Wyner-Ziv Codec for Video. In: *SPIE Visual Communications and Image Processing (VCIP)*. Santa Clara, CA, USA, 2004

Aaron u. a. 2004b

AARON, Anne ; RANE, Shantenu ; GIROD, Bernd: Wyner-Ziv Video Coding with Hash-based Motion Compensation at the Receiver. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Singapore, October 2004

Aaron u. Girod 2004

AARON, Anne ; GIROD, Bernd: Wyner-Ziv Video Coding with low Encoder Complexity. In: *Proc. Picture Coding Symposium (PCS)*. SF/USA, December 2004

Aaron u. a. 2006

AARON, Anne ; VARODAYAN, David ; GIROD, Bernd: Wyner-Ziv Residual Coding of Video. In: *Proc. Picture Coding Symposium (PCS)*. Beijing, China, 2006

Adikari u. a. 2006

ADIKARI, A.B.B. ; FERNANDO, W.A.C. ; WEERAKKODY, W.A.R.J. ; ARACHCHI, H.K.: A Sequential Motion Compensation Refinement Technique for Distributed Video Coding of Wyner-Ziv Frames. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Atlanta, GA, USA, 2006

Adikari u. a. 2007

ADIKARI, A.B.B. ; FERNANDO, W.A.C. ; WEERAKKODY, W.A.R.J.: Iterative Wyner-Ziv Decoding for Unidirectional Distributed Video Coding. In: *Electronics Letters* 43 (2007), 18, Nr. 2, S. 93 –95. – ISSN 0013–5194

Alparone u. a. 1996

ALPARONE, L. ; BARNI, M. ; BARTOLINI, F. ; CAPPELLINI, V.: Adaptively Weighted Vector-Median Filters for Motion-Fields Smoothing. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Georgia, USA, 1996

Artigas u. Torres 2005a

ARTIGAS, X. ; TORRES, L.: Iterative generation of motion-compensated side information for distributed video coding. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Genova, Italy, 2005

Artigas u. a. 2007a

ARTIGAS, X. ; ASCENSO, J. ; DALAI, M. ; KLOMP, S. ; KUBASOV, D. ; OUARET, M.: The DISCOVER Codec: Architecture, Techniques and Evaluation. In: *Proc. Picture Coding Symposium (PCS)*. Lisboa, Portugal, 2007

Artigas u. Torres 2006

ARTIGAS, Xavi ; TORRES, Luis: An Approach to Distributed Video Coding

using 3D Face Models. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Atlanta, GA, USA, 2006

Artigas u. a. 2007b

ARTIGAS, Xavi ; MALINOWSKI, Simon ; GUILLEMOT, Christine ; TORRES, Luis: Overlapped Quasi-Arithmetic Codes for Distributed Video Coding. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. San Antonio, TX, USA, 2007

Artigas u. Torres 2005b

ARTIGAS, Xavier ; TORRES, Luis: A Model-based Enhanced Approach to Distributed Video Coding. In: *Proc. Image Analysis for Multimedia Interactive Services (WIAMIS)*. Montreux, Switzerland, 2005

Ascenso u. a. 2005

ASCENSO, João ; BRITES, Catarina ; PEREIRA, Fernando: Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding. In: *Proc. Conference on Speech and Image Processing, Multimedia Communications and Services (EURASIP)*. Slovak Republic, 2005

Ascenso u. Pereira 2007

ASCENSO, João ; PEREIRA, Fernando: Adaptive Hash-Based Side Information Exploitation for Efficient Wyner-Ziv Video Coding. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. San Antonio, TX, USA, 2007

Ascenso u. Pereira 2008

ASCENSO, João ; PEREIRA, Fernando: Hierarchical Motion Estimation for Side Information Creation in Wyner-Ziv Video Coding. In: *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*. New York, NY, USA : ACM, 2008. – ISBN 978–1–59593–993–7

Ash 1990

ASH, Robert B.: *Information Theory*. New York : Dover Publication, Inc., 1990

Astola u. a. 1990

ASTOLA, J. ; HAAVISTO, P. ; NEUVO, Y.: Vector Median Filters. In: *Proceedings of the IEEE* 78 (1990), april, Nr. 4, S. 678 –689. – ISSN 0018–9219

Badem u. a. 2010

BADEM, M. ; FERNANDO, W.A.C. ; KONDOZ, A.M.: Unidirectional Distributed

Video Coding Using Dynamic Parity Allocation and Improved Reconstruction. In: *5th International Conference on Information and Automation for Sustainability (ICIAFs)*. Sri Lanka, 2010

Bahl u. a. 1974

BAHL, L. ; COCKE, J. ; JELINEK, F. ; RAVIV, J.: Optimal decoding of linear codes for minimizing symbol error rate (Corresp.). In: *IEEE Transactions on Information Theory* 20 (1974), march, Nr. 2, S. 284 – 287. – ISSN 0018-9448

Baraniuk 2007

BARANIUK, R.G.: Compressive Sensing [Lecture Notes]. In: *IEEE Signal Processing Magazine* 24 (2007), july, Nr. 4, S. 118 –121. – ISSN 1053-5888

Bay u. a. 2008

BAY, Herbert ; ESS, Andreas ; TUYTELAARS, Tinne ; GOOL, Luc V.: SURF: Speeded Up Robust Features. In: *Computer Vision and Image Understanding (CVIU)* 110 (2008), Nr. 3, S. 346–359

de Berg u. a. 2008

BERG, Mark de ; CHEONG, Otfried ; KREVELD, Marc van ; OVERMARS, Mark: *Computational Geometry - Algorithms and Applications*. 3rd. Berlin : Springer, 2008

Berrou u. a. 1993

BERROU, C. ; GLAVIEUX, A. ; THITIMAJSHIMA, P.: Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. In: *IEEE International Conference on Communications (ICC)*. Geneva, Switzerland, 1993

Bertalmio u. a. 2000

BERTALMIO, M. ; SAPIRO, G. ; CASELLES, V. ; BALLESTER, C.: Image Inpainting. In: *Proc. SIGGRAPH*. New Orleans, USA, 2000

Borchert u. a. 2007

BORCHERT, S. ; WESTERLAKEN, R.P. ; GUNNEWIEK, R. K. ; LAGENDIJK, R.L.: On Extrapolating Side Information in Distributed Video Coding. In: *Proc. Picture Coding Symposium (PCS)*. Lisboa, Portugal, 2007

Borchert u. a. 2008

BORCHERT, S. ; WESTERLAKEN, R.P. ; GUNNEWIEK, R.K. ; LAGENDIJK, R.L.:

Motion compensated prediction in transform domain Distributed Video Coding. In: *IEEE 10th Workshop on Multimedia Signal Processing (MMSP)*. Cairns, Australia, 2008

Borchert 2010

BORCHERT, Stefan: *Distributed Video Coding (DVC): Motion estimation and DCT quantization in low complexity video compression*, ASCI graduate school, Diss., 2010

Bossert 1998

BOSSERT, Martin: *Kanalcodierung*. Bd. 2. Auflage. Stuttgart : B.G. Teubner, 1998

Brites u. a. 2006a

BRITES, Catarina ; ASCENSO, João ; PEREIRA, Fernando: Improving Transform Domain Wyner-Ziv Video Coding Performance. In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Toulouse, France, 2006

Brites u. a. 2006b

BRITES, Catarina ; ASCENSO, João ; PEREIRA, Fernando: Studying Temporal Correlation Noise Modeling for Pixel Based Wyner-Ziv Video Coding. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Atlanta, GA, USA, 2006

Brites u. Pereira 2007

BRITES, Catarina ; PEREIRA, Fernando: Encoder Rate Control for Transform Domain Wyner-Ziv Video Coding. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. San Antonio, TX, USA, 2007

Brites u. Pereira 2011

BRITES, Catarina ; PEREIRA, Fernando: An Efficient Encoder Rate Control Solution for Transform Domain Wyner-Ziv Video Coding. In: *IEEE Transactions on Circuits and Systems for Video Technology* 21 (2011), september, Nr. 9, S. 1278 –1292. – ISSN 1051–8215

Bronstein u. a. 2001

BRONSTEIN, I. N. ; SEMENDJAJEW, K. A. ; MUSIOL, G. ; MÜHLIG, H.: *Taschenbuch der Mathematik*. Frankfurt am Main : Verlag Harri Deutsch, 2001

Bross u. a. 2012

BROSS, B. ; HAN, W.-J. ; OHM, J.-R. ; SULLIVAN, G.J. ; WIEGAND, T.: High Efficiency Video Coding (HEVC) Text Specification Draft 7 / JCT-VC. 2012 (JCTVC-I1003). – Forschungsbericht

Candes u. Wakin 2008

CANDES, E.J. ; WAKIN, M.B.: An Introduction To Compressive Sampling. In: *IEEE Signal Processing Magazine* 25 (2008), march, Nr. 2, S. 21 –30. – ISSN 1053–5888

Chen u. a. 2008

CHEN, David ; VARODAYAN, David ; FLIERL, Markus ; GIROD, Bernd: Wyner-ZIV Coding of Multiview Images with Unsupervised Learning of Disparity and Gray Code. In: *Proc. 15th IEEE International Conference on Image Processing ICIP 2008*. San Diego, CA, USA, 2008

Chen u. a. 2010

CHEN, Hung-Wei ; KANG, Li-Wei ; LU, Chun-Shien: Dictionary Learning-based Distributed Compressive Video Sensing. In: *Picture Coding Symposium (PCS)*. Nagoya, Japan, 2010

Cover u. Thomas 1991

COVER, Thomas M. ; THOMAS, Joy A.: *Elements of Information Theory*. New York : Wiley, 1991

Deligiannis u. a. 2011

DELIGIANNIS, N. ; VERBIST, F. ; BARBARIEN, J. ; SLOWACK, J. ; WALLE, R. Van d. ; SCHELKENS, P. ; MUNTEANU, A.: Distributed Coding of Endoscopic Video. In: *IEEE International Conference on Image Processing (ICIP)*. Brussels, Belgium, 2011

DISCOVER 2007

DISCOVER Distributed Coding for Video Services. <http://www.discoverdvc.org/>, november 2007. – zuletzt abgerufen: 05.01.2012

Do u. a. 2009

DO, T.T. ; CHEN, Yi ; NGUYEN, D.T. ; NGUYEN, N. ; GAN, Lu ; TRAN, T.D.: Distributed Compressed Video Sensing. In: *IEEE International Conference on Image Processing (ICIP)*. Cairo, Egypt, 2009

Dragotti u. Gastpar 2009

DRAGOTTI, Pier L. ; GASTPAR, Michael: *Distributed Source Coding - Theory, Algorithms, and Applications*. Burlington : Academic Press, 2009

Dufaux u. a. 2009

DUFAUX, Frederic ; GAO, Wen ; TUBARO, Stefano ; VETRO, Anthony: Distributed Video Coding: Trends and Perspectives. In: *EURASIP Journal on Image and Video Processing* 2009 (2009), S. 13

Dufaux u. Ebrahimi 2010

DUFAUX, Frederic ; EBRAHIMI, Touradj: Encoder and Decoder Side Global and Local Motion Estimation for Distributed Video Coding. In: *IEEE International Workshop on Multimedia Signal Processing (MMSP)*. Saint-Malo, France, 2010

Feig u. Winograd 1992

FEIG, E. ; WINOGRAD, S.: Fast Algorithms for the Discrete Cosine Transform. In: *IEEE Transactions on Signal Processing* 40 (1992), september, Nr. 9, S. 2174 –2193. – ISSN 1053–587X

Fleming u. a. 2004

FLEMING, M. ; ZHAO, Qian ; EFFROS, M.: Network Vector Quantization. In: *IEEE Transactions on Information Theory* 50 (2004), august, Nr. 8, S. 1584 – 1604. – ISSN 0018–9448

Flierl u. a. 1998

FLIERL, M. ; WIEGAND, T. ; GIROD, B.: A Locally Optimal Design Algorithm for Block-based Multi-Hypothesis Motion-Compensated Prediction. In: *Proc. Data Compression Conference (DCC)*. Snowbird, Utah, USA, 1998

Gallager 1962

GALLAGER, Robert G.: Low-Density Parity-Check Codes. In: *IRE Transactions on Information Theory* 8 (1962), january, Nr. 1, S. 21 –28. – ISSN 0096–1000

Gallager 1963

GALLAGER, Robert G.: *Low Density Parity Check Codes*. Cambridge : M.I.T. Press, 1963

Garcia-Frias u. Zhao 2001

GARCIA-FRIAS, Javier ; ZHAO, Ying: Compression of Correlated Binary

Sources Using Turbo Codes. In: *IEEE Communications Letters* 5 (2001), october, Nr. 10, S. 417–419

Girod u. a. 2005

GIROD, B. ; AARON, A.M. ; RANE, S. ; REBOLLO-MONEDERO, D.: Distributed Video Coding. In: *Proceedings of the IEEE* 93 (2005), january, Nr. 1, S. 71–83

Göbel 2007

GÖBEL, Jürgen: *Informationtheorie und Codierungsverfahren*. Berlin : VDE Verlag, 2007

Goyal u. a. 2008

GOYAL, V.K. ; FLETCHER, A.K. ; RANGAN, S.: Compressive Sampling and Lossy Compression. In: *IEEE Signal Processing Magazine* 25 (2008), march, Nr. 2, S. 48 –56. – ISSN 1053–5888

Guo u. a. 2006a

GUO, Mei ; LU, Yan ; WU, Feng ; ZHAO, Debin ; GAO, Wen: Practical Wyner-Ziv Switching Scheme for Multiple Bit-Rate Video Streaming. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Atlanta, GA, USA, 2006

Guo u. a. 2006b

GUO, Xun ; LU, Yan ; WU, Feng ; GAO, Wen: Distributed video coding using wavelet. In: *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*. Island of Kos, Greece, 2006

H.264 2012

H.264 Referenzsoftware. <http://iphome.hhi.de/suehring/tml/download/>, 2012. – zuletzt abgerufen: 16.08.2012

Hagenauer 1988

HAGENAUER, J.: Rate-Compatible Punctured Convolutional Codes (RCPC codes) and their Applications. In: *IEEE Transactions on Communications* 36 (1988), april, Nr. 4, S. 389 –400. – ISSN 0090–6778

Hänsel u. Müller 2009a

HÄNSEL, R. ; MÜLLER, E.: Error Locating for plausible Wyner-Ziv Video Coding using Turbo Codes. In: *Proc. of the 2009 IEEE International Workshop on Multimedia Signal Processing (MMSP)*. Rio de Janeiro, Brazil, 2009

Hänsel u. Müller 2009b

HÄNSEL, R. ; MÜLLER, E.: Improved Reconstruction for Distributed Video Coding. In: *Proc. International Conference on Ultra Modern Communications (ICUMT)*. St.-Petersburg, Russia, 2009

Hänsel u. Müller 2009c

HÄNSEL, R. ; MÜLLER, E.: Simplifying the Decoder Rate Control Scheme for Distributed Video Coding by Flexible Slepian-Wolf Decoding. In: *In Proc. Pacific-Rim Symposium on Image and Video Technology (PSIVT)*. Tokyo, Japan, 2009

Hänsel u. Müller 2010

HÄNSEL, R. ; MÜLLER, E.: Confidence Features Extraction for Wyner-Ziv Video Decoding. In: *Proc. European Signal Processing Conference (EUSIPCO)*. Aalborg, Denmark, 2010

Hänsler 2001

HÄNSLER, Eberhard: *Statistische Signale*. Berlin : Springer, 2001

Harris u. Stephens 1988

HARRIS, C. ; STEPHENS, M.: A Combined Corner and Edge Detection. In: *Proceedings of Alvey Vision Conference*. Manchester, England, 1988

HEVC 2012

HEVC Testmodelsoftware. <https://hevc.hhi.fraunhofer.de/>, 2012. – zuletzt abgerufen: 16.08.2012

Höher 2011

HÖHER, Peter A.: *Grundlagen der digitalen Informationsübertragung*. 1. Wiesbaden : Vieweg + Teubner, 2011

Horn u. Schunck 1981

HORN, Berthold K. P. ; SCHUNCK, Brian G.: Determining Optical Flow. In: *Artificial Intelligence* 17 (1981), S. 185–203

Huang u. a. 2009

HUANG, Xin ; BRITES, C. ; ASCENSO, J. ; PEREIRA, F. ; FORCHHAMMER, S.: Distributed Video Coding with Multiple Side Information. In: *Proc. Picture Coding Symposium (PCS)*. Chicago, IL, USA, 2009

Kitakami u. Sano 2005

KITAKAMI, M. ; SANO, J.: Code Design and Decoding Methods for Burst Error Locating Codes. In: *Proc. Pacific Rim International Symposium on Dependable Computing (PRDC)*. Changsha, Hunan, China, 2005

Kodavalla u. Mohan 2011

KODAVALLA, V.K. ; MOHAN, P.G.K.: Chroma Components Coding in Feedback-Free Distributed Video Coding. In: *Proc. IEEE International Conference on Internet Multimedia Systems Architecture and Application (IMSAA)*. Bangalore, India, 2011

Kubasov u. a. 2007a

KUBASOV, D. ; LAJNEF, K. ; GUILLEMOT, C.: A hybrid Encoder/Decoder Rate Control for a Wyner-Ziv Video Codec with a Feedback Channel. In: *IEEE MultiMedia Signal Processing Workshop*. Chania, Crete, 2007

Kubasov u. Guillemot 2006

KUBASOV, Denis ; GUILLEMOT, Christine: Mesh-Based Motion-Compensated Interpolation for Side Information Extraction in Distributed Video Coding. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Atlanta, GA, USA, 2006

Kubasov u. a. 2007b

KUBASOV, Denis ; NAYAK, Jayanth ; GUILLEMOT, Christine: Optimal Reconstruction in Wyner-Ziv Video Coding with Multiple Side Information. In: *Proc. IEEE Workshop on Multimedia Signal Processing (MMSP)*. Chania, Crete, Greece, 2007

Kühn 2006

KÜHN, Volker: *Wireless Communications over MIMO Channels - Application to CDMA and Multiple Antenna Systems*. Chichester : John Wiley & Sons, Ltd, 2006

Kuo u. a. 2012

KUO, Yonghong ; ZHANG, Peng ; CHEN, Jian: Feedback Free Multiple Description Distributed Video Coding for Robust Transmission. In: *Electronics Letters* 48 (2012), 7, Nr. 12, S. 691 –692. – ISSN 0013–5194

Lee 1984

LEE, Byeong: A New Algorithm to Compute the Discrete Cosine Transform. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 32 (1984), december, Nr. 6, S. 1243 – 1245. – ISSN 0096–3518

Lee u. Huang 1994

LEE, PeiZong ; HUANG, Fang-Yu: Restructured Recursive DCT and DST Algorithms. In: *IEEE Transactions on Signal Processing* 42 (1994), july, Nr. 7, S. 1600 –1609. – ISSN 1053–587X

Li u. a. 2004

LI, J. ; TU, Z. ; BLUM, R.S.: Slepian-Wolf Coding for Nonuniform Sources Using Turbo Codes. In: *Data Compression Conference (DCC)*. Snowbird, Utah, USA, 2004

Li u. a. 2010

LI, Zhu ; TOMIOKA, Y. ; KITAZAWA, H.: Motion Estimation Based on Iterative Color Matching and Structure Matching. In: *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. Kuala Lumpur, Malaysia, 2010

Lin u. a. 2007

LIN, Yao-Chung ; VARODAYAN, David ; ; GIROD, Bernd: Image Authentication and Tampering Localization using Distributed Source Coding. In: *Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP)*. Crete, Greece, 2007

Lin u. a. 2008

LIN, Yao-Chung ; VARODAYAN, D. ; FINK, T. ; BETTERS, E. ; GIROD, B.: Localization of Tampering in Contrast and Brightness Adjusted Images Using Distributed Source Coding and Expectation Maximization. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. San Diego, CA, USA, 2008

Liu u. a. 2006

LIU, Limin ; LI, Zhen ; DELP, Edward J.: Backward Channel Aware Wyner-Ziv Video Coding. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Atlanta, GA, USA, 2006

Liu u. a. 2007

LIU, Limin ; LI, Zhen ; ; DELP, Edward J.: Complexity-Rate-Distortion Analysis of Backward Channel Aware Wyner-Ziv Video Coding. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. San Antonio, TX, USA, 2007

Loeffler u. a. 1989

LOEFFLER, C. ; LIGTENBERG, A. ; MOSCHYTZ, G.S.: Practical Fast 1-D DCT Algorithms with 11 Multiplications. In: *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Seattle, WA, USA, 1989

Lowe 2004

LOWE, David G.: Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision* 60 (2004), S. 91–110. – ISSN 0920–5691

Luong u. a. 2011

LUONG, Huynh V. ; HUANG, Xin ; FORCHHAMMER, S.: Parallel Iterative Decoding of Transform Domain Wyner-Ziv Video Using Cross Bitplane Correlation. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Brussels, Belgium, 2011

MacKay 2005

MACKAY, David J.: *Information Theory, Inference, and Learning Algorithms*. Cambridge : Cambridge University Press, 2005

Magnus u. Heudecker 2007

MAGNUS, Jan R. ; HEUDECKER, Heinz: *Matrix Differential Calculus – with Applications in Statistics and Econometrics*. 3. Chichester : John Wiley & Sons, Ltd, 2007

Maitre u. a. 2006

MAITRE, Matthieu ; GUILLEMOT, Christine ; MORIN, Luce: 3D Scene Modeling for Distributed Video Coding. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Atlanta, GA, USA, 2006

Maitre u. a. 2007

MAITRE, Matthieu ; GUILLEMOT, Christine ; MORIN, Luce: 3-D Model-Based

Frame Interpolation for Distributed Video Coding of Static Scenes. In: *IEEE Transactions on Image Processing* 16 (2007), May, Nr. 5, S. 1246–1257

Malinowski u. a. 2009

MALINOWSKI, S. ; ARTIGAS, X. ; GUILLEMOT, C. ; TORRES, L.: Distributed Coding Using Punctured Quasi-Arithmetic Codes for Memory and Memoryless Sources. In: *Proc. Picture Coding Symposium (PCS)*. Chicago, IL, USA, 2009

Malvar u. a. 2003

MALVAR, H.S. ; HALLAPURO, A. ; KARCZEWICZ, M. ; KEROFISKY, L.: Low-Complexity Transform and Quantization in H.264/AVC. In: *IEEE Transactions on Circuits and Systems for Video Technology* 13 (2003), July, Nr. 7, S. 598 – 603. – ISSN 1051–8215

Marcellin u. a. 2000

MARCELLIN, M.W. ; GORMISH, M.J. ; BILGIN, A. ; BOLIEK, M.P.: An Overview of JPEG-2000. In: *Proc. Data Compression Conference (DCC)*. Snowbird, Utah, USA, 2000

Martinez u. a. 2008a

MARTINEZ, J.L. ; HOLDER, C. ; FERNANDEZ, G.E. ; KALVA, H. ; QUILLES, F.: DVC Using a Half-Feedback Based Approach. In: *Proc. IEEE International Conference on Multimedia and Expo (ICME)*. Hannover, Germany, 2008

Martinez u. a. 2008b

MARTINEZ, J.L. ; FERNANDEZ-ESCRIBANO, G. ; KALVA, H. ; WEERAKKODY, W.A.R.J. ; FERNANDO, W.A.C. ; GARRIDO, A.: Feedback Free DVC Architecture Using Machine Learning. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. San Diego, CA, USA, 2008

Martins u. a. 2009

MARTINS, R. ; BRITES, C. ; ASCENSO, J. ; PEREIRA, F.: Refining Side Information for Improved Transform Domain Wyner-Ziv Video Coding. In: *IEEE Transactions on Circuits and Systems for Video Technology* 19 (2009), September, Nr. 9, S. 1327 –1341. – ISSN 1051–8215

Matas u. a. 2004

MATAS, J ; CHUM, O ; URBAN, M ; PAJDLA, T: Robust Wide-Baseline Stereo

from Maximally Stable Extremal Regions. In: *Image and Vision Computing* 22 (2004), Nr. 10, S. 761 – 767. – ISSN 0262–8856

Meng u. a. 2011

MENG, Jin ; YANG, En-Hui ; HE, Da-Ke: Linear Interactive Encoding and Decoding for Lossless Source Coding With Decoder Only Side Information. In: *IEEE Transactions on Information Theory* 57 (2011), august, Nr. 8, S. 5281 –5297. – ISSN 0018–9448

Meur 2011

MEUR, Olivier L.: Video Compression Beyond H.264, HEVC / IRISA. 2011. – Forschungsbericht

Mikolajczyk u. Schmid 2005

MIKOLAJCZYK, K. ; SCHMID, C.: A Performance Evaluation of Local Descriptors. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005), october, Nr. 10, S. 1615 –1630. – ISSN 0162–8828

Moon 2005

MOON, Todd K.: *Error Correction Coding - Mathematical Methods and Algorithms*. Chichester : John Wiley & Sons, Ltd, 2005

Morbée u. a. 2007

MORBEE, M. ; PRADES-NEBOT, J. ; PIZURICA, A. ; PHILIPS, W.: Rate Allocation Algorithm for Pixel-Domain Distributed Video Coding Without Feedback Channel. In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. Honolulu, HI, USA, 2007

Moreira u. Farrell 2006

MOREIRA, Jorge C. ; FARRELL, Patrick G.: *Essentials of Error-Control Coding*. Chichester : John Wiley & Sons, Ltd, 2006

Natário u. a. 2005

NATÁRIO, Luís ; BRITES, Catarina ; ASCENSO, João ; PEREIRA, Fernando: Extrapolating Side Information for Low-Delay Pixel-Domain Distributed Video Coding. In: *Proc. International Workshop on Very Low Bitrate Video (VLBV)*. Sardinia, Italy, 2005

Ohm 2004

OHM, Jens-Rainer: *Multimedia Communication Technology*. Berlin : Springer, 2004

Park u. a. 2012

PARK, Sang-Uk ; LEE, Young-Yoon ; KIM, Chang-Su ; LEE, Sang-Uk: Efficient Side Information Generation Using Assistant Pixels for Distributed Video Coding. In: *Proc. Picture Coding Symposium (PCS)*. Krakow, Poland, 2012

Pereira u. a. 2006

PEREIRA, F. ; (IST), P. C. ; ACOSTA, E. ; (UPC), L.Torres ; (INRIA), C. G. ; OUARET, M. ; DUFAUX, F. ; (EPFL), T. E. ; LEONARDI, R. ; (UNIBS), M. D. ; (UH), S. K.: Application Scenarios and Functionalities for DVC - Version 1 / DISCOVER DVC. Version: 2006. <http://www.discoverdvc.org/deliverables/Discover-D4.pdf>. 2006. – Forschungsbericht

Pereira u. a. 2008

PEREIRA, Fernando ; TORRES, Luis ; GUILLEMOT, Christine ; EBRAHIMI, Touradj ; LEONARDI, Riccardo ; KLOMP, Sven: Distributed Video Coding: Selecting the Most Promising Application Scenarios. In: *Signal Processing: Image Communication* 23 (2008), S. 339–352

Petrazzuoli u. a. 2010a

PETRAZZUOLI, Giovanni ; CAGNAZZO, Marco ; PESQUET-POPESCU, Béatrice: Fast and Efficient Side Information Generation in Distributed Video Coding by Using Dense Motion Representations. In: *Proc. European Signal Processing Conference (EUSIPCO)*. Aalborg, Denmark, 2010

Petrazzuoli u. a. 2010b

PETRAZZUOLI, Giovanni ; CAGNAZZO, Marco ; PESQUET-POPESCU, Béatrice: High Order Motion Interpolation for Side Information Improvement in DVC. In: *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Dallas, TX, USA, 2010

Prades-Nebot u. a. 2009

PRADES-NEBOT, J. ; MA, Yi ; HUANG, T.: Distributed Video Coding using Compressive Sampling. In: *Proc. Picture Coding Symposium (PCS)*. Chicago, IL, USA, 2009

Pradhan u. Ramchandran 1999

PRADHAN, S. S. ; RAMCHANDRAN, Kannan: Distributed Source Coding Using Syndromes (DISCUS): Design and Construction. In: *Proc. IEEE Data Compression Conference (DCC)*. Snowbird, Utah, USA, 1999

Priziment u. Malah 2008

PRIZIMENT, Y. ; MALAH, D.: Model-based Rate Allocation in Distributed Video Coding Systems. In: *Proc. IEEE Convention of Electrical and Electronics Engineers in Israel (IEEEI)*, 2008

Puri u. a. 2006

PURI, R. ; MAJUMBAR, A. ; ISHWAR, P. ; RAMCHANDRAN, K.: Distributed Video Coding in Wireless Sensor Networks. In: *IEEE Signal Processing Magazine* 23 (2006), July, Nr. 4, S. 94–106

Puri u. Ramchandran 2002

PURI, Rohit ; RAMCHANDRAN, Kannan: PRISM: A New Video Coding Architecture Based on Distributed Compression Principles / EECS Department, University of California, Berkeley. 2002. – Forschungsbericht

Puri u. Ramchandran 2003

PURI, Rohit ; RAMCHANDRAN, Kannan: PRISM: A Video Coding Paradigm Based on Motion-Compensated Prediction at the Decoder / EECS Department, University of California, Berkeley. 2003. – Forschungsbericht

Puri u. a. 2007

PURI, Rohit ; MAJUMDAR, Abhik ; RAMCHANDRAN, Kannan: PRISM: A Video Coding Paradigm With Motion Estimation at the Decoder. In: *IEEE Transactions on Image Processing* 16 (2007), october, Nr. 10, S. 2436–2448

Ramamoorthy u. a. 2006

RAMAMOORTHY, A. ; JAIN, K. ; CHOU, P.A. ; EFFROS, M.: Separating Distributed Source Coding from Network Coding. In: *IEEE Transactions on Information Theory* 52 (2006), june, Nr. 6, S. 2785 – 2795. – ISSN 0018–9448

Rane u. a. 2005

RANE, S. ; AARON, A. ; GIROD, B.: Error-Resilient Video Transmission Using Multiple Embedded Wyner-Ziv Descriptions. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Genova, Italy, 2005

Rane u. a. 2004

RANE, Shantanu ; AARON, Anne ; GIROD, Bernd: Systematic Lossy Forward Error Protection for Error-Resilient Digital Video Broadcasting - A Wyner-Ziv Coding Approach. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Singapore, 2004

Richardson 2003

RICHARDSON, Iain E.: *H.264 and MPEG-4 Video Compression - Video Coding for Next-generation Multimedia*. Chichester : John Wiley & Sons, Ltd, 2003

Richter 2006

RICHTER, Henryk: *Standardübergreifende Konzepte für die blockbasierte Videocodierung*, Fakultät für Informatik und Elektrotechnik der Universität Rostock, Diss., 2006

Rosten u. Drummond 2005

ROSTEN, E. ; DRUMMOND, T.: Fusing Points and Lines for High Performance Tracking. In: *Proc. IEEE International Conference on Computer Vision (ICCV)*. Beijing, China, 2005

Rosten u. a. 2005

ROSTEN, Edward ; REITMAYR, Gerhard ; DRUMMOND, Tom: Real-Time Video Annotations for Augmented Reality. In: *Proc. International Symposium on Visual Computing. (ISVC)*. Lake Tahoe, NV, USA, 2005

Rosten u. Drummond 2006

ROSTEN, Edward ; DRUMMOND, Tom: Machine Learning for High-Speed Corner Detection. In: *Proc. European Conference on Computer Vision (ECCV)*. Graz, Austria, 2006

Rowitch u. Milstein 2000

ROWITCH, D.N. ; MILSTEIN, L.B.: On The Performance of Hybrid FEC/ARQ Systems Using Rate Compatible Punctured Turbo (RCPT) Codes. In: *IEEE Transactions on Communications* 48 (2000), June, Nr. 6, S. 948–959

Said 1995

SAID, A.: Error Burst Detection with High-Rate Convolutional Codes. In: *Proc. IEEE International Symposium on Information Theory*. Whistler, Canada, 1995

Seiler u. Kaup 2008

SEILER, Jürgen ; KAUP, André: Adaptive Joint Spatio-Temporal Error Concealment for Video Communication. In: *Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP)*. Cairns, Queensland, Australia, 2008

Shannon 1948

SHANNON, Claude E.: A Mathematical Theorie of Communications. In: *Bell System Technical Journal* 27 (1948), S. 379–423

Shannon u. Weaver 1949

SHANNON, Claude E. ; WEAVER, Warren: *The Mathematical Theory of Communication*. Chicago : University of Illinois Press, 1949

Shokrollahi 2003

SHOKROLLAHI, Amin: LDPC Codes: An Introduction / Digital Fountain, Inc. 39141 Civic Center Drive, Fremont, CA 94538, April 2 2003. – Forschungsbericht

Skorupa u. a. 2009

SKORUPA, J. ; SLOWACK, J. ; MYS, S. ; LAMBERT, P. ; WALLE, R. Van d. ; GRECOS, C.: Stopping Criteria for Turbo Coding in a Wyner-Ziv Video Codec. In: *Picture Coding Symposium (PCS)*. 2009 Chicago, IL, USA, 2009

Slepian u. Wolf 1973

SLEPIAN, D. ; WOLF, J.: Noiseless Coding of Correlated Information Sources. In: *IEEE Transactions on Information Theory* 19 (1973), July, Nr. 4, S. 471–480

Slowack u. a. 2012a

SLOWACK, J. ; SKORUPA, J. ; DELIGIANNIS, N. ; LAMBERT, P. ; MUNTEANU, A. ; WALLE, R. Van d.: Distributed Video Coding With Feedback Channel Constraints. In: *IEEE Transactions on Circuits and Systems for Video Technology* 22 (2012), july, Nr. 7, S. 1014 –1026. – ISSN 1051–8215

Slowack u. a. 2012b

SLOWACK, J. ; DELIGIANNIS, N. ; LAMBERT, P. ; MUNTEANU, A. ; WALLE, R. Van d.: Feedback-Constrained Wyner-Ziv Video Coding. In: *Proc. Picture Coding Symposium (PCS)*. Krakow, Poland, 2012

Smolić u. a. 1999

SMOLIĆ, A. ; SIKORA, T. ; OHM, J.-R.: Long-term Global Motion Estimation

and Its Application for Sprite Coding, Content Description, and Segmentation. In: *IEEE Transactions on Circuits and Systems for Video Technology* 9 (1999), december, Nr. 8, S. 1227 –1242. – ISSN 1051–8215

Smolić 2001

SMOLIĆ, Aljoscha: *Globale Bewegungsbeschreibung und Video Mosaiking unter Verwendung Parametrischer 2-D Modelle, Schätzverfahren und Anwendungen*, RWTH Aachen, Diss., 2001

Sofke 2009

SOFKE, Sören: *Fehlerverschleierung für Verteilte Videocodierung*, Universität Rostock - Institut für Nachrichtentechnik, Diplomarbeit, Januar 2009

Sofke u. a. 2009

SOFKE, Sören ; HÄNSEL, Ralph ; MÜLLER, Erika: Human Visual System Aware Decoding Strategies for Distributed Video Coding. In: *Proc. Picture Coding Symposium (PCS)*. 2009 Chicago, IL, USA, 2009

Steinmüller 2008

STEINMÜLLER, Johannes: *Bildanalyse*. Berlin : Springer, 2008

Strutz 2002

STRUTZ, Tilo: *Bilddatenkompression - Grundlagen, Codierung, JPEG, MPEG, Wavelets*. 2. Wiesbaden : Vieweg, 2002

Sullivan u. a. 2006

SULLIVAN, Gary J. ; OHM, Jens-Rainer ; ORTEGA, Antonio ; DELP, Edward ; VETRO, Anthony ; BARNI, Mauro: Future of Video Coding and Transmission. In: *IEEE Signal Processing Magazine* Volume 23, Issue 6 (2006), november, S. 76–82

Sullivan u. Baker 1991

SULLIVAN, G.J. ; BAKER, R.L.: Motion Compensation for Video Compression Using Control Grid Interpolation. In: *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Toronto, Ontario, Canada, 1991

Sun u. a. 2006

SUN, Wen ; LU, Yan ; WU, Feng: Bit-Stream Switching in Multiple Bit-Rate

Video Streaming Using Wyner-Ziv Coding. In: *Proc. International Conference on Multimedia and Expo (ICME)*. Toronto, Canada, 2006

Tagliasacchi u. a. 2006

TAGLIASACCHI, M. ; TRAPANESE, A. ; TUBARO, S. ; ASCENSO, J. ; BRITES, C. ; PEREIRA, F.: Exploiting Spatial Redundancy in Pixel Domain Wyner-Ziv Video Coding. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Atlanta, GA, USA, 2006

Tomé u. Pereira 2011

TOMÉ, António ; PEREIRA, Fernando: Low Delay Distributed Video Coding with Refined Side Information. In: *Signal Processing: Image Communication* 26 (2011), Nr. 4-5, S. 220 – 235. – ISSN 0923–5965

Toto-Zarasoia u. a. 2009

TOTO-ZARASOIA, V. ; MAGLI, E. ; ROUMY, A. ; OLMO, G.: On Distributed Arithmetic Codes and Syndrome Based Turbo Codes for Slepian-Wolf Coding of Non Uniform Sources. In: *Proc. European Signal Processing Conference (EU-SIPCO)*. Glasgow, Scotland, 2009

Trapanese u. a. 2005

TRAPANESE, Alan ; TAGLIASACCHI, Marco ; TUBARO, Stefano ; ASCENSO, João ; BRITES, Catarina ; PEREIRA, Fernando: Embedding a Block-based Intra Mode in Frame-based Pixel Domain Wyner-Ziv Video Coding. In: *Proc. International Workshop on Very Low Bitrate Video (VLBV)*. Sardinia, Italy, 2005

Tschumperlé 2002

TSCHUMPERLÉ, David: *PDE's Based Regularization of Multivalued Images and Applications*, Ecole Doctorale Sciences et Technologies de l'Information et de la Communication, Diss., 2002

Ungerboeck 1987

UNGERBOECK, G.: Trellis-Coded Modulation with Redundant Signal Sets Part I: Introduction. In: *IEEE Communications Magazine* 25 (1987), february, Nr. 2, S. 5 –11. – ISSN 0163–6804

Varodayan u. a. 2005

VARODAYAN, David ; AARON, Anne ; GIROD, Bernd: Rate-Adaptive Distribu-

ted Source Coding using Low-Density Parity-Check Codes. In: *Proc. Asilomar Conference on Signals, Systems, and Computers*. Pacific Grove, CA, USA, 2005

Varodayan u. a. 2006

VARODAYAN, David ; AARON, Anne ; GIROD, Bernd: Rate-Adaptive Codes for Distributed Source Coding. In: *EURASIP Signal Processing* 86 (2006), Nr. 11, S. 3123–3130

Varodayan u. a. 2007

VARODAYAN, David ; LIN, Cao-Chung ; MAVLANKAR, Aditya ; FLIERL, Markus ; GIROD, Bernd: Wyner-Ziv Coding of Stereo Images with Unsupervised Learning of Disparity. In: *Proc. Picture Coding Symposium (PCS)*. Lisboa, Portugal, 2007

Varodayan u. Girod 2010

VARODAYAN, David ; GIROD, Bernd: Side-Information-Adaptive Distributed Source Coding. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Hong Kong, 2010

Vatis u. a. 2007

VATIS, Yuris ; KLOMP, Sven ; OSTERMANN, Jörn: Inverse Bit Plane Decoding Order for Turbo Code Based Distributed Video Coding. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. San Antonio, TX, USA, 2007

Vedaldi u. Fulkerson 2012

VEDALDI, A. ; FULKERSON, B.: *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. <http://www.vlfeat.org/>, february 2012. – zuletzt abgerufen: 29.02.2012

Wallace 1992

WALLACE, G.K.: The JPEG Still Picture Compression Standard. In: *IEEE Transactions on Consumer Electronics* 38 (1992), february, Nr. 1, S. xviii – xxxiv. – ISSN 0098–3063

Wang u. Ortega 2004

WANG, Huisheng ; ORTEGA, A.: Scalable Predictive Coding by Nested Quantization with Layered Side Information. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Singapore, 2004

Wang u. a. 2004

WANG, Huisheng ; CHEUNG, Ngai-Man ; ORTEGA, Antonio: WZS: Wyner-Ziv Scalable Predictive Video Coding. In: *Proc. Picture Coding Symposium (PCS)*. San Francisco, USA, 2004

Wang u. a. 2006

WANG, Huisheng ; CHEUNG, Ngai-Man ; ORTEGA, Antonio: A Framework for Adaptive Scalable Video Coding Using Wyner-Ziv Techniques. In: *EURASIP Journal on Applied Signal Processing* 2006 (2006), S. 18 pages

Westerlaken u. a. 2006

WESTERLAKEN, Ronald P. ; BORCHERT, Stefan ; GUNNEWIEK, Rene K. ; LAGENDIJK, R.(Inald) L.: Finding a Near Optimal Dependency Channel Model for a LDPC-based Wyner-Ziv Video Compression Scheme. In: *Proc. Annual Conference of the Advanced School for Computing and Imaging*. Lommel, Belgium, 2006

Westerlaken u. a. 2007

WESTERLAKEN, Ronald P. ; BORCHERT, Stefan ; GUNNEWIEK, Rene K. ; LAGENDIJK, R.(Inald) L.: Analyzing Symbol and Bit Plane-Based LDPC in Distributed Video Coding. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. San Antonio, TX, USA, 2007

Wiegand u. a. 2003

WIEGAND, T. ; SULLIVAN, G.J. ; BJONTEGAARD, G. ; LUTHRA, A.: Overview of The H.264/AVC Video Coding Standard. In: *IEEE Transactions on Circuits and Systems for Video Technology* 13 (2003), july, Nr. 7, S. 560 –576. – ISSN 1051–8215

Wyner u. Ziv 1976

WYNER, Aaron D. ; ZIV, Jacob: The Rate-Distortion Function for Source Coding with Side Information at The Decoder. In: *IEEE Transactions on Information Theory* 22 (1976), January, Nr. 1, S. 1–10

Yaacoub u. Farah 2009

YAACOUB, C. ; FARAH, J.: An Information-Theoretic Approach for Removing Feedback Channels in Distributed Video Coding. In: *Proc. International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)*. Zouk Mosbeh, Libanon, 2009

Yang u. He 2008

YANG, En-Hui ; HE, Da-Ke: On Interactive Encoding and Decoding for Lossless Source Coding with Decoder Only Side Information. In: *Proc. IEEE International Symposium on Information Theory (ISIT)*. Toronto, ON, Canada, 2008

Yang u. He 2010

YANG, En-Hui ; HE, Da-Ke: Interactive Encoding and Decoding for One Way Learning: Near Lossless Recovery With Side Information at the Decoder. In: *IEEE Transactions on Information Theory* 56 (2010), april, Nr. 4, S. 1808 – 1824. – ISSN 0018–9448

Yang u. a. 2006

YANG, Feng ; DING, Guiguang ; DAI, Qionghai ; YIN, Yaguang: Wavelet Based Scalable Wyner Ziv Video Coding. In: *Proc. Picture Coding Symposium (PCS)*. Beijing, China, 2006

Ye u. a. 2008

YE, Shuiming ; OUARET, M. ; DUFAUX, F. ; EBRAHIMI, T.: Hybrid Spatial and Temporal Error Concealment for Distributed Video Coding. In: *Proc. IEEE International Conference on Multimedia and Expo (ICME)*. Hannover, Germany, 2008

Ye u. a. 2009

YE, Shuiming ; OUARET, Mourad ; DUFAUX, Frederic ; EBRAHIMI, Touradj: Improved Side Information Generation for Distributed Video Coding by Exploiting Spatial and Temporal Correlations. In: *EURASIP Journal on Image and Video Processing* 2009 (2009), Nr. 1, S. 683510. – ISSN 1687–5281

Zhao u. Garcia-Frias 2002

ZHAO, Ying ; GARCIA-FRIAS, Javier: Data Compression of Correlated Non-Binary Sources Using Punctured Turbo Codes. In: *Proc. IEEE Data Compression Conference (DCC)*. Snowbird, Utah, USA, 2002. – ISBN 0–7695–1477–4

Anhang



Vertiefende Ausführungen zur Kanalcodierung und Bildverarbeitung

Dieses Kapitel beinhaltet vertiefende Ausführungen zu den Bereichen der Kanalcodierung und Bildverarbeitung. Dies umfasst die LDPC Codes (Abschn. A.1.1) und CRC Codes (Abschn. A.1.2), welche für die Umsetzung der Verteilten Videocodierung hohe Relevanz besitzen. Weiterhin werden ausführliche Darstellungen zu den Bewegungsmodellen und der Bewegungsschätzung (Abschn. A.2.1) als auch zu der Bestimmung von markanten Bereichen in einem Bild gegeben (Abschn. A.2.2).

A.1. Kanalcodierung

A.1.1. LDPC - Low Density Parity Check

LDPC (*Low Density Parity Check*) Codes wurden von Robert G. Gallager entwickelt [Gallager 1962, 1963] und besitzen sehr gute Fehlerkorrektureigenschaften. Diese Codes gehören auch zu den linearen Blockcodes und können durch Generator- \mathbf{G} und Prüfmatrix \mathbf{H} dargestellt werden. Die besondere Eigenschaft dieser Codes ist die dünn besetzte Prüfmatrix.

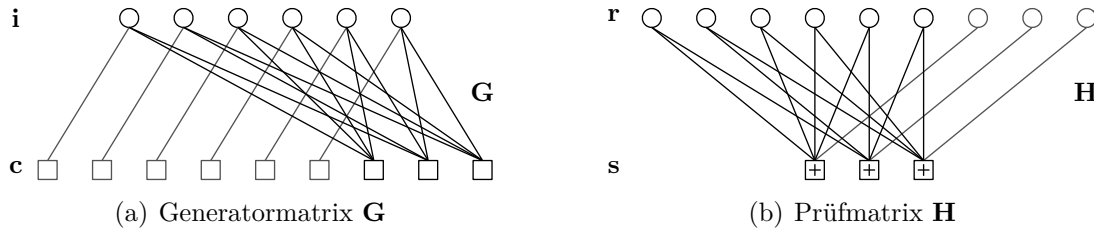


Abbildung A.1.: Darstellung der Generator- und Prüfmatrix eines LDPC Codes als Tannergraph

Beide Matrizen können als Tanner-Graph (Abb. A.1) dargestellt werden [Shokrollahi 2003]. Der bipartite Graph zur Encodierung enthält zum einen die Quellknoten (*source nodes*), welche den Informationsvektor \mathbf{i} widerspiegeln. Der Codevektor \mathbf{c} wird durch modulo 2 Addition der durch Kanten verbundenen Quellknoten bestimmt. In diesem Beispiel handelt es sich um einen systematischen LDPC Code mit $k = 6$ und einer Codewortlänge $n = 9$. Der zugehörige nichtsystematische Teil \mathbf{A} der Generator- und Prüfmatrix ist in Gleichung A.1 dargestellt (Abb. A.1, schwarz hervorgehoben).

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} \quad (\text{A.1})$$

Der zur Prüfmatrix \mathbf{H} korrespondierende Tanner-Graph beinhaltet den empfangenen Codevektor \mathbf{r} als Quellknoten. Die Prüfknotten \mathbf{s} (Syndrom) ergeben für ein gültiges Codewort den Nullvektor $\mathbf{0}$. Die Anzahl der Kanten an jedem Knoten r_i gibt den Grad des Codes an. Hierbei unterscheidet man zwischen regulären und irregulären Codes. Beim regulären Fall ist der Grad für alle Knoten gleich. Irreguläre Codes besitzen Knoten unterschiedlichen Grades und zeigen eine höhere Leistungsfähigkeit als reguläre LDPC Codes.

Die Komplexität der Encodierung für systematische LDPC Codes ist gering, da nur $n - k$ Redundanzknoten (*redundant node*) bestimmt werden müssen. Die Anzahl der Modulo-2-Additionen n_{add} für ein Codewort in Abhängigkeit vom durchschnittlichen Grad \bar{v} ergibt sich wie folgt: $n_{add} = \bar{v} \cdot k - (n - k)$.

Die Decodierung eines LDPC Codes erfolgt meist über den *sum-product algo-*

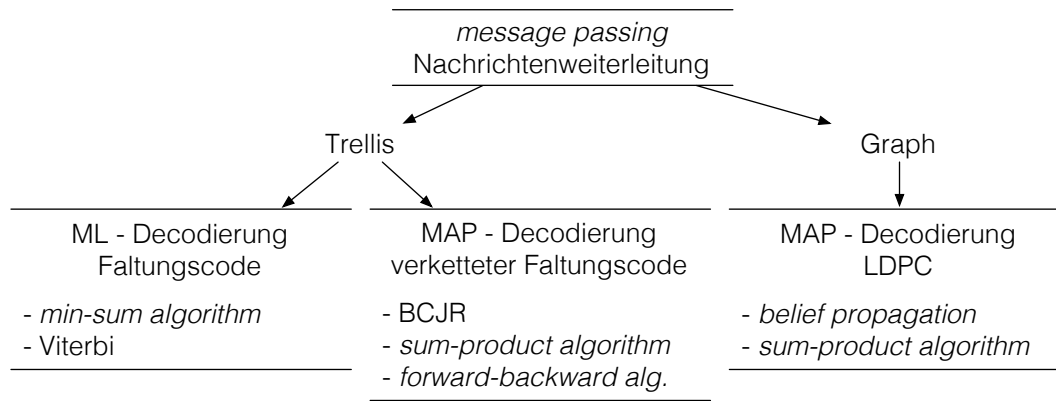


Abbildung A.2.: Decodierungsverfahren für Faltungs- und LDPC Codes

rithm (Summen-Produkt-Methode), welcher sich als sehr leistungsfähig herausgestellt hat [MacKay 2005, S.557][Kühn 2006]. Diese Methode wird auch *belief propagation* genannt und gehört zur Gruppe der *message passing* Verfahren auf Graphen (Abb. A.2). Die Decodierung stellt hierbei nicht sicher, dass ein gültiges Codewort $\hat{\mathbf{c}}$ erzeugt wird, vielmehr wird für jedes Symbol \hat{c}_j der wahrscheinlichste Wert unter Bedingung des Empfangsvektors \mathbf{r} bzw. \mathbf{y} gewählt (Maximierung von $\Pr(\hat{\mathbf{c}}|\mathbf{r})$ bzw. $\Pr(\hat{\mathbf{c}}|\mathbf{y})$). Damit handelt es sich hier um eine symbolweise Maximum-a-posteriori (MAP) Decodierung. Ein ungültiges Codewort stellt hierbei kein Decodierungsversagen dar, da im Falle eines systematischen Codes der Informationsvektor $\hat{\mathbf{i}}$ direkt aus dem Codevektor $\hat{\mathbf{c}}$ ermittelt werden kann [Bossert 1998, S.15].

A.1.2. CRC - Cyclic Redundancy Check

Die bisher vorgestellten (verketteten) Faltungs- oder LDPC Codes dienen im Wesentlichen der Fehlerkorrektur. Andererseits können Kanalcodes auch eingesetzt werden, um Übertragungsfehler zu erkennen. Ein weitverbreiteter Vertreter der Fehlererkennungs_codes ist der CRC Code [Moreira u. Farrell 2006, S.92] [Göbel 2007, S.206]. Hierbei handelt es sich auch um einen linearen zyklischen Code.

Die **Encodierung** erfolgt durch Polynomdivision. Dabei werden die Elemente des Informationsvektors \mathbf{i} als binäre Koeffizienten eines Polynoms $P(x)$ interpretiert. Das Codewort wird aus dem systematischen Anteil \mathbf{c}_i und den Prüfsymbolen \mathbf{c}_p

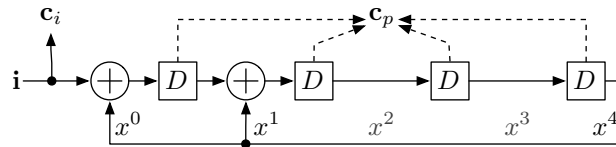


Abbildung A.3.: Schieberegisterstruktur zur Generierung des CRC Codewortes
(CRC-4, $G(x) = x^4 + x + 1$)

zusammengesetzt, wobei die Prüfsymbole durch Polynomdivision erzeugt werden (Gl. A.2).

$$E(x) = \text{mod}(P(x), G(x)) \quad (\text{A.2})$$

$G(x)$ bezeichnet das Generatorpolynom, welches charakteristisch für den Code ist und maßgeblich die Leistungsfähigkeit bestimmt. Die Prüfsymbole entsprechen dem Fehlerpolynom $E(x)$, welches dem Rest der Polynomdivision $P(x)$ durch $G(x)$ entspricht. Die Restdivision kann effizient durch die Verwendung einer Schieberegisterstruktur umgesetzt werden (Abb. A.3, [Moon 2005, S.131]). Nachdem der gesamte Informationsvektor in der Schieberegisterstruktur verarbeitet wurde, kann der Divisionsrest aus den Schieberegistern ausgelesen werden.

Die **Decodierung** beziehungsweise Überprüfung erfolgt, indem der empfangene Informationsvektor $\hat{\mathbf{i}}$ erneut codiert und anschließend die so erzeugen Prüfsymbole mit den übertragenden verglichen werden.

A.2. Bildverarbeitung

A.2.1. Bewegungsschätzung

Die Bewegungsschätzung nimmt eine zentrale Rolle in der Bildsequenzcodierung ein. Mit der Schätzung der Bewegung ist es möglich, eine gute zeitliche Prädiktion des Bildinhaltes durch Bewegungskompensation zu erzeugen und somit die Leistungsfähigkeit des Videocodierungssystems wesentlich zu steigern. Im Bereich der

Verteilten Videocodierung ist die Bestimmung der Bewegung auch von zentraler Bedeutung, da die Seiteninformation meist durch bewegungskompensierte temporale Interpolation erzeugt wird.

„Ziel der Bewegungsschätzung ist es, die Bewegungsparameter bezüglich des verwendeten Bewegungsmodells zu bestimmen“ [Smolić 2001, S.23]. Damit kann der Prozess der Bewegungsschätzung in die wesentlichen Bereiche „Bewegungsmodelle“ und „Methoden zur Bestimmung der Bewegungsparameter“ zerlegt werden. Bevor verschiedene Modelle und Methoden vorgestellt werden, wird eine Klassifikation der in Videosequenzen auftretenden Bewegungsarten vorgenommen.

Klassifikation der Bewegungsarten in Videosequenzen

Bewegungen in Videosequenzen werden durch Veränderungen in der von der Kamera betrachteten Szene hervorgerufen. Die Kamera kann dabei ausschließlich Änderungen des Grauwertes beobachten. Ändert sich der Grauwert nicht, so kann keine Bewegung bestimmt werden. Der Grauwert kann sich aber auch dann ändern, wenn keine Bewegung stattgefunden hat. Im Wesentlichen unterscheidet man in drei Bewegungsarten:

Kamerabewegungen — Durch die Bewegung der Kamera (Verschiebung, Schwenk, Rollen, Zoom) verändern sich alle Objekte im Bild gleichmäßig. Dabei treten Bildveränderungen wie **Translation**, **Rotation**, **Skalierung** und **perspektivische Verzerrung** auf.

bewegte Objekte — Bewegte Objekte bewirken ähnliche Änderungen wie diese, die durch die Kamerabewegung entstehen, solche Veränderungen sind jedoch auf einzelne Bildbereiche beschränkt. Darüber hinaus können **Auf-** und **Verdeckungen** an den Objektgrenzen auftreten sowie ein Objekt durch **Deformation** im Erscheinungsbild verändert werden.

scheinbare Bewegungen — Neben den durch Kamerabewegung oder bewegte Objekte hervorgerufenen Änderungen in der Szene können auch scheinbare Bewegungen, zum Beispiel durch **Änderungen der Beleuchtung** oder **Störeinflüsse** (Rauschen, Artefakte), auftreten. Verschiebt sich zum Beispiel der Grauwertgradient auf einer gekrümmten Fläche in Folge einer Änderung des Einfallswinkels des Lichtes, so erscheint dies wie eine Bewegung des Objektes.

Bewegungsmodelle

Die Bewegung in einer Szene muss durch ein Modell erfasst werden, um eine Bewegungskompensation durchführen zu können. Da die Bestimmung eines Modells der gesamten dreidimensionalen Szene sehr aufwendig und fehleranfällig ist, werden im Allgemeinen hinreichend genaue vereinfachte Modelle verwendet. Diese lassen sich in drei Gruppen einteilen:

globales Bewegungsmodell — Ein globales Bewegungsmodell beschreibt die Bewegung des gesamten Bildes mit wenigen Parametern. Ein Beispiel ist das 8-parametrische Modell (Gl. A.3), welches sowohl Translation (a_1, b_1) , Skalierung, Scherung und Rotation (a_2, a_3, b_2, b_3) als auch perspektivische Verzerrungen (c_1, c_2) abbilden kann [Smolić 2001][Smolić u. a. 1999]. In der Verteilten Videocodierung wurde dies in [Dufaux u. Ebrahimi 2010] im Rahmen der Seiteninformationsermittlung eingesetzt.

$$x' = \frac{a_1 + a_2x + a_3y}{1 + c_1x + c_2y} \quad y' = \frac{b_1 + b_2x + b_3y}{1 + c_1x + c_2y} \quad (\text{A.3})$$

Bewegungen einzelner Objekte sowie Auf- und Verdeckungen können mit diesem Modell nicht abgebildet werden.

Pixel-, blockbasiertes Bewegungsmodell — Pixel- bzw. blockbasierte Bewegungsmodelle weisen jedem Pixel bzw. Block einen Bewegungsvektor zu. Es wird davon ausgegangen, dass benachbarte Pixel die gleiche Bewegung aufweisen. Für jeden Pixel bzw. Block wird eine translatorische Bewegung abgebildet [Ohm 2004][Strutz 2002]. Diese Verfahren sind in der klassischen Videocodierung weit verbreitet [Richardson 2003][Wiegand u. a. 2003]. In Abhängigkeit von der Blockgröße (Pixel $\cong 1 \times 1$ Block) kann die Bewegung fein bzw. grob aufgelöst werden. Wird eine sehr feine Auflösung gewählt, so können komplexe Bewegungen (Rotation, Verzerrung) durch ein dichtes Bewegungsvektorfeld angenähert werden. Diskontinuitäten der Bewegung können bei hinreichend kleinen Blockgrößen näherungsweise abgebildet werden. In der Verteilten Videocodierung wird dieses Modell unter anderem in [Ascenso u. Pereira 2008][Ascenso u. a. 2005] angewendet.

Netzmodell — Komplexe Deformationen von Bildbereichen, wie zum Beispiel das Gesicht eines Nachrichtensprechers, können mit einem Netzmodell (*Warping*,

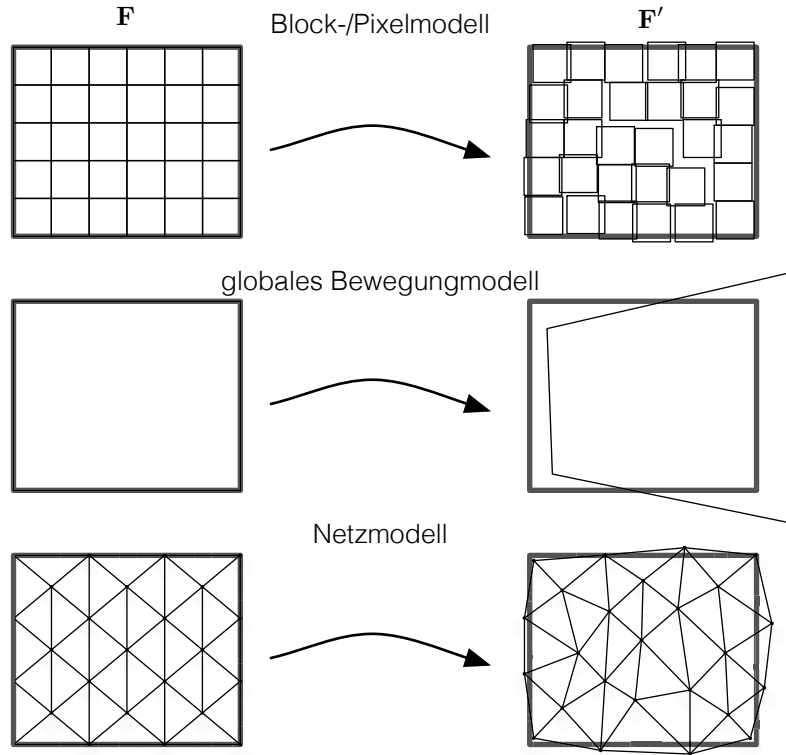


Abbildung A.4.: Modelle zur Beschreibung von Bewegungen in Videosequenzen

(*Control Grid, Mesh-based*) abgebildet werden. Hierbei wird jedem Knotenpunkt eine Bewegung zugeordnet und für den dazwischenliegenden Bildbereich interpoliert [Ohm 2004][Smolić 2001][Strutz 2002][Kubasov u. Guillemot 2006]. Dieses Modell kann keine diskontinuierlichen Bewegungen, wie sie zum Beispiel an Objektgrenzen auftreten, abbilden.

Methoden zur Bestimmung der Bewegungsmodellparameter

Die verwendete Methode zur Bestimmung der Parameter \mathbf{mp} des Bewegungsmodells hängt direkt von dem verwendeten Modell ab (Abb. A.5). Der Ansatz besteht dabei zumeist in der Minimierung des Fehlers zwischen Originalbild \mathbf{F} und dem bewegungskompensierten Referenzbild \mathbf{F}'^{MVF} (Gl. A.4).

$$\mathbf{mp} = \arg \min_{\mathbf{mp}} \sum_{x,y} \left| \mathbf{F}(x,y) - \mathbf{F}'(\text{model}(x,y,\mathbf{mp})) \right|^\alpha \quad (\text{A.4})$$

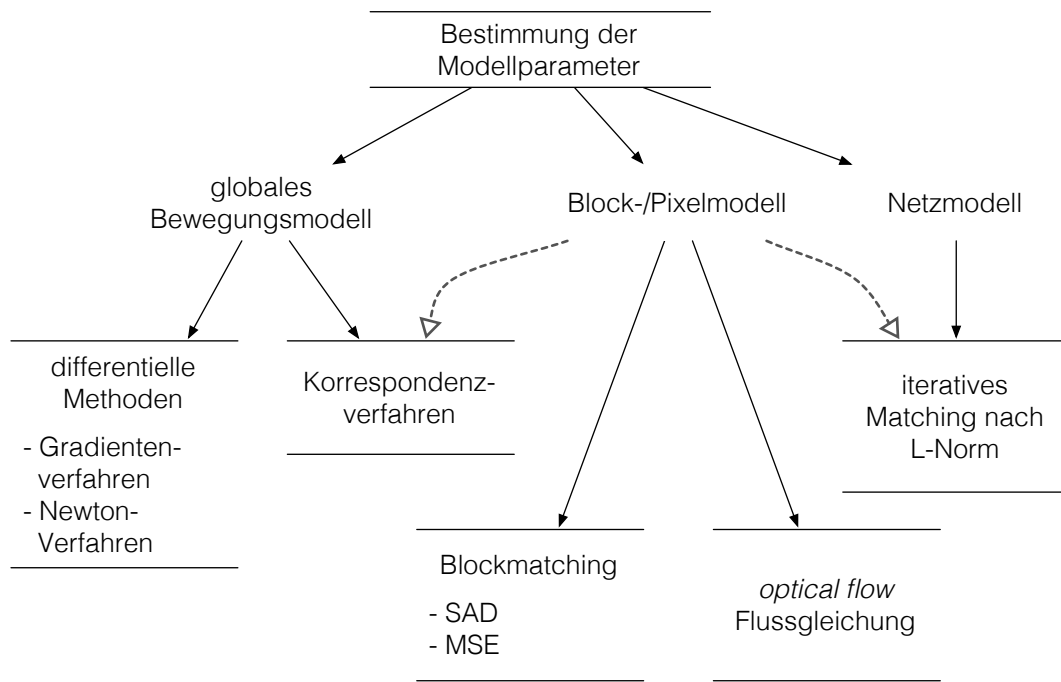


Abbildung A.5.: Bestimmung der Modellparameter zur Beschreibung von Bewegungen in Videosequenzen

Die Funktion *model* bildet die Koordinaten im Originalbild auf die im Referenzbild unter Berücksichtigung der Modellparameter **mp** ab. Weiterhin wird meist der mittlere lineare ($\alpha = 1$) oder mittlere quadratische Fehler ($\alpha = 2$) berücksichtigt.

Im Folgenden werden einige übliche Verfahren zur Bestimmung der Modellparameter kurz beschrieben.

Blockmatching

Blockmatching ist ein einfaches weitverbreitetes Verfahren zur Bestimmung der Parameter des block- bzw. pixelbasierten Bewegungsmodells. Jedem Block bzw. Pixel wird dabei ein Bewegungsvektor **mv** zugewiesen, indem für die Umgebung *MW* jedes Blocks/Pixels im Originalbild **F** die entsprechende Umgebung im Referenzbild **F'** gesucht wird. Maßgebend ist die Minimierung des Fehlermaßes (Gl. A.5). Verbreitet ist hier die Verwendung des mittleren quadratischen (*MSE*, $\alpha = 2$) oder auch linearen Abstandes (*SAD*, $\alpha = 1$). Die Suche wird meist auf den Suchbereich *SW* des Referenzbildes eingeschränkt.

$$\begin{aligned} \mathbf{mv} = \arg \min_{\mathbf{mv}} \sum_{(k,l) \in MW} & |\mathbf{F}(x_0 + k, y_0 + l) \dots \\ & \dots - \mathbf{F}'(x_0 + k + mv_x, y_0 + l + mv_y)|^p \end{aligned} \quad (\text{A.5})$$

$$\mathbf{mv} \in SW = \{(mv_x, mv_y) \in \mathbb{Q}_r^2\} \quad (\text{A.6})$$

$$\mathbf{mv} = \mathbf{MVF}(x_0, y_0) \quad (\text{A.7})$$

$$MW = \{(k, l) \in \mathbb{Q}_r^2\} \quad (\text{A.8})$$

$$\mathbb{Q}_b = \left\{ q \in \mathbb{Q} \mid \dots - \frac{2}{b}; -\frac{1}{b}; 0; \frac{1}{b}; \frac{2}{b}; \dots \right\} \quad (\text{A.9})$$

Dieses Verfahren liefert Bewegungsvektoren in der Schrittweite von einem Pixel (*full-pel*, $r = 1$). Ist eine höhere Genauigkeit gefordert, so kann dies durch entsprechende Interpolation des Original- und des Referenzbildes erreicht werden. Üblich sind hier Halb-, Viertel- oder Achtel-Pixel-Genauigkeit ($r = 2$, $r = 4$ oder $r = 8$).

Flussgleichung (optical flow)

Der optische Fluss beschreibt die Änderung des Grauwertes zwischen Bildern einer Videosequenz [Steinmüller 2008]. Treten keine scheinbaren Bewegungen auf, so können aus dem optischen Fluss die Bewegungsparameter ermittelt werden. Die Parameter (u, v) für jeden einzelnen Pixel können auf Basis der Flussgleichung (Gl. A.10) bestimmt werden. Im Fall von Videosequenzen, welche diskrete Orts- und Zeitdimensionen aufweisen, werden die Differenzialquotienten durch die entsprechenden Differenzenquotienten ersetzt.

$$\frac{\partial F}{\partial x}u + \frac{\partial F}{\partial y}v + \frac{\partial F}{\partial t} = 0 \quad (\text{A.10})$$

In der Literatur werden verschiedene Verfahren zur Lösung der Flussgleichung beschrieben [Ohm 2004][Horn u. Schunck 1981][Steinmüller 2008]. Bei diesen ist es notwendig, dass eine zusätzliche Nebenbedingung eingeführt wird, da das Gleichungssystem A.10 unterbestimmt ist. Als Nebenbedingung kann zum Beispiel die Glattheit des optischen Flusses $\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2$ und $\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2$ verwendet werden [Horn u. Schunck 1981].

Weiterhin darf der Gradient in x- als auch in y-Richtung nicht verschwinden, um eine Lösung zu erhalten. Dies wird als Aperturbedingung bezeichnet. Vereinfacht ausgedrückt muss die Umgebung Gradienten in mehr als nur einer Richtung aufweisen, um eine Bewegungsschätzung durchführen zu können. Dies trifft auch auf das Blockmatching zu.

Die Bestimmung der Bewegungsparameter über die optische Flussgleichung liefert implizit Sub-Pixel genaue Bewegungsparameter. Starke Bewegungen können nur so weit bestimmt werden, wie die Grauwertgradienten konstant sind, da die Flussgleichung auf den linearen Gliedern einer Taylor-Reihenentwicklung basiert.

Netzmodell

Die Parameter des Netzmodells [Sullivan u. Baker 1991][Ohm 2004] können mittels eines iterativen Verfahrens bestimmt werden. Dabei werden nacheinander für jeden Knotenpunkt die Bewegungsparameter so gewählt, dass der Fehler in dessen Einflussbereich minimal wird. Da die Fehlerfunktion von den Parametern der anderen Knotenpunkte beeinflusst wird, sind hier mehrere Iterationen notwendig. Dies kann beschleunigt werden, indem gute Startwerte gewählt werden, welche durch ein vorheriges Blockmatching durch Verwendung des Blockmodells erzeugt wurden.

Globales Bewegungsmodell

Für die Bestimmung der Parameter eines globalen Bewegungsmodells gibt es zwei Gruppen von Verfahren. Zum einen differentielle Methoden, welche die Parameter nach dem Gradientenverfahren oder Newton-Verfahren so anpassen, dass der Fehler zwischen Originalbild und bewegungskompensiertem Referenzbild minimal wird. Zum anderen kann in einem zweistufigen Prozess zuerst die Bewegung einzelner Bildbereiche bestimmt und anschließend das Modell an diese vorläufigen Bewegungsparameter angepasst werden. Hierbei spricht man von Korrespondenzverfahren [Smolić 2001], wobei die Bewegungsvektoren eines Blockmodells als Korrespondenz dienen können.

Probleme bei der Bestimmung von Bewegungsmodellparametern

Bei der Bestimmung der Parameter eines Bewegungsmodells treten im Allgemeinen folgende Probleme auf:

lokales Minimum — Die Minimierung der Fehlerfunktion durch ein iteratives Verfahren kann in einem lokalen Minimum enden, womit nicht die optimalen Parameter des Bewegungsmodells ermittelt werden können. Dieses Problem kann auch beim einfachen Blockmatching auftreten, wenn der Suchraum SW zu klein gewählt und somit das globale Minimum nicht erfasst wird. Beide beschriebenen Probleme können durch die Bestimmung guter Startwerte umgangen werden.

reale Bewegung — Durch die Minimierung der Fehlerfunktion können scheinbare Bewegungen abgebildet werden, welche nicht der realen Bewegung entsprechen.

Gradienten — Bildbereiche, die keinen Gradienten im Grauwert aufweisen (z.B. homogene Flächen), eignen sich nicht für die Bewegungsschätzung und können zum Beispiel im Fall des Blockmodells zu lokalen Störungen des Bewegungsvektorfeldes führen.

Auf- und Verdeckungen — Bildbereiche, die im Originalbild aber nicht im Referenzbild sichtbar sind, führen zu unterschiedlichen Problemen bei der Bestimmung der Modellparameter. Zum einen kann z.B. beim Blockmatching keine Korrespondenz gefunden werden. Zum anderen treten Auf- und Verdeckungen an Objektgrenzen und somit Sprünge im Bewegungsfeld auf, welche durch die Bewegungsmodelle zum Teil nicht abgedeckt werden können (globales Bewegungsmodell, Netzmodell)

A.2.2. Merkmalsbereiche in Bildern

Die Erkennung gleicher Objekte in unterschiedlichen Bildern ist eine wichtige Aufgabe der Bildverarbeitung. Der menschliche Betrachter bewältigt solche Aufgaben, indem charakteristische Merkmale wie Form, Farbe oder Struktur verglichen werden. Dabei kann der Mensch Beleuchtungsunterschiede und Verzerrungen durch 3-dimensionale Transformation, wie zum Beispiel Rotationen, ausgleichen.

In der maschinellen Bildverarbeitung bedient man sich dazu charakteristischer Bildbereiche, welche miteinander verglichen werden, um gleiche Objekte zu finden. Allgemein kann man diesen Prozess in drei Stufen unterteilen:

Detektion (Detection) — Finden von Bildbereichen, die charakteristisch für dieses Objekt sind (Repräsentant, *feature point*), eine gewisse Eindeutigkeit (Einzigartigkeit) ist hilfreich

Merkmalsbestimmung (Feature Extraction) — Bestimmung einer charakteristischen Beschreibung (Merkmalsvektor) des Repräsentanten, z.B. aus der Struktur der Umgebung

Zuordnung (Matching) — Zuordnung ähnlicher Repräsentanten in anderen Bildern oder in großen Datenbanken, basierend auf dem Merkmalsvektor

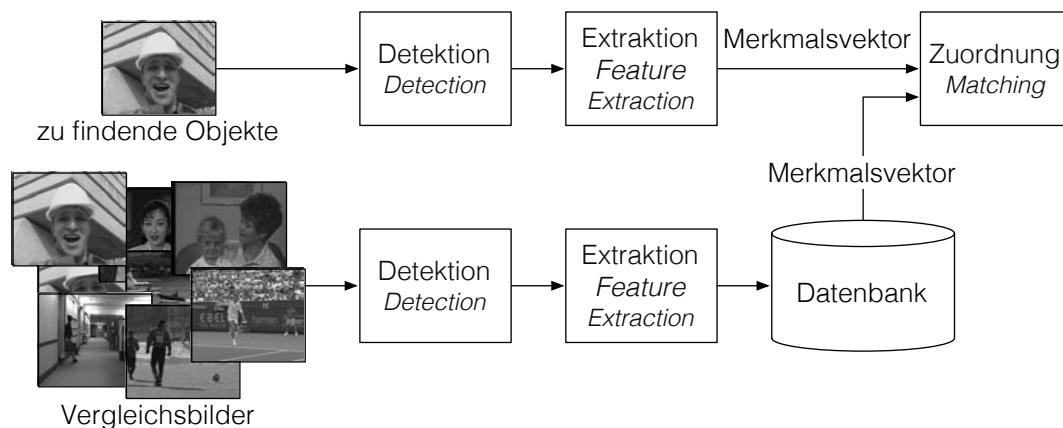


Abbildung A.6.: Arbeitsablauf zum Finden von Objekten in einer Menge von Bildern anhand charakteristischer Bildmerkmale

Betrachtet man den Anwendungsfall, Objekte in einer großen Menge von Bildern zu finden, so kann der in Abbildung A.6 dargestellte Arbeitsablauf angewendet werden. Es werden in der Vorbereitungsphase Repräsentanten in den Vergleichsbildern bestimmt (Detektion), wobei im Weiteren zu jedem Repräsentanten ein Merkmalsvektor extrahiert (Extraktion) und in einer Datenbank abgelegt wird. Sollen nun einzelne Objekte diesen Vergleichsbildern zugeordnet werden, wird auch hier eine Detektion und Extraktion durchgeführt. Diese Merkmalspunkte werden dann anhand des Merkmalsvektors zugeordnet. Diese Herangehensweise reduziert die Komplexität, da nur die kompakte Beschreibung durch Merkmalsvektoren zugeordnet werden muss.

Weiterhin wird die Detektion, Extraktion und Zuordnung markanter Bildbereiche auch eingesetzt, um der Bewegung von Objekten zu folgen [Li u. a. 2010] oder auch,

um die dreidimensionale Struktur einer Szene in einem Stereokamerasystem (*Depth Estimation*, [Matas u. a. 2004]) zu bestimmen.

Die aufgeführten Anwendungsbeispiele können nur bei hoher Robustheit der Zuordnung (*Matching*) von Merkmalspunkten umgesetzt werden. Aus diesem Grund ist es die Zielsetzung, dass Detektion und Extraktion robust gegenüber

- Änderung der Beleuchtung,
- Deformation,
- dreidimensionale Transformation und
- teilweise Verdeckung

sind. Dies wird je nach Verfahren mit unterschiedlichen Mitteln erreicht. Das als sehr leistungsfähig bekannte Verfahren SIFT (*Scale Invariant Feature Transform*, [Lowe 2004]) wird auf Seite 209 im Detail vorgestellt.

Ein wichtiger Detektor ist der *Harris Corner Detector* [Harris u. Stephens 1988], welcher auf der Auswertung der Hesse-Matrix (*Hessian Matrix*) basiert. Für mobile Anwendungen wurde der FAST Detektor entwickelt [Rosten u. Drummond 2005; Rosten u. a. 2005; Rosten u. Drummond 2006], weiterhin wird für Stereoanwendungen zum Beispiel der MSER (*maximally stable extremal regions*, [Matas u. a. 2004]) eingesetzt.

Wichtige Deskriptoren sind beispielsweise das weitverbreitete SIFT-Verfahren, GLOH (*Gradient Location and Orientation Histogram*, [Mikolajczyk u. Schmid 2005]) oder der auf Geschwindigkeit optimierte SURF Deskriptor (*Speeded-Up Robust Features*, [Bay u. a. 2008]).

Harris Detektor (*Harris Corner Detector*)

Der Harris Detektor [Harris u. Stephens 1988] wurde entworfen, um Bildbereiche in die drei Klassen Kante, Ecke und Fläche einzuteilen. Dieser Detektor kann somit charakteristische Bereiche eines Bildes bestimmen. Da dieses Verfahren ausschließlich auf den Gradienten basiert, ist es somit auch robust gegenüber Helligkeitsänderungen, jedoch nicht gegenüber Kontraständerungen.

Basis für die Detektion von Ecken im Bild stellt die Betrachtung der Fehlerfunktion E (quadratischer Fehler, Gl. A.11) für kleine Verschiebungen in der Umgebung eines Merkmalspunktes dar.

$$E(u, v) = \sum_{x,y} \mathbf{G} \mathbf{W}_\sigma(x, y) \{ \mathbf{F}(x, y) - \mathbf{F}(x + u, y + v) \} \quad (\text{A.11})$$

Besitzt die Fehlerfunktion $E(u, v)$ ein ausgeprägtes scharfes Minimum, so handelt es sich um einen Eckpunkt. Ein ausgeprägtes scharfes Minimum ist dann gegeben, wenn die Krümmung in Haupt- und Nebenrichtung stark ist (*principal curvature*, 2. Ableitung). Kanten, beziehungsweise flache Bereiche, sind durch eine in nur einer Richtung stark gekrümmte, beziehungsweise flache Fehlerfunktion gekennzeichnet.

Die Krümmung in Haupt- und Nebenrichtung der Fehlerfunktion E wird unter Verwendung der Hesse-Matrix (*Hessian Matrix*, Gl. A.12, [Magnus u. Heudecker 2007, S.114]) bestimmt, wobei die Eigenwerte α, β der Matrix \mathbf{H} die Krümmung in Haupt- und Nebenrichtung angeben.

$$\mathbf{H}(E) = \begin{pmatrix} \frac{\partial^2 E}{\partial u^2} & \frac{\partial^2 E}{\partial u \partial v} \\ \frac{\partial^2 E}{\partial v \partial u} & \frac{\partial^2 E}{\partial v^2} \end{pmatrix} \quad (\text{A.12})$$

In [Harris u. Stephens 1988] wird zur Vereinfachung angenommen, dass die Grauwerte des Bildes \mathbf{F} in der Umgebung eines Merkmalspunktes einer schiefen Ebene entsprechen und somit die Fehlerfunktion einer zweidimensionalen Parabel entspricht. Damit ergibt sich für die Hesse-Matrix die vereinfachte Schreibweise nach Gleichung A.15, wobei die ersten Ableitungen der Grauwerte des Bildes \mathbf{F} mittels des Operators $\mathbf{d} = (-1, 0, 1)$ approximiert werden.

$$\mathbf{F}_x = \mathbf{F} * \mathbf{d} \approx \frac{\partial F}{\partial x} \quad (\text{A.13})$$

$$\mathbf{F}_y = \mathbf{F} * \mathbf{d}^T \approx \frac{\partial F}{\partial y} \quad (\text{A.14})$$

$$\mathbf{H}(E) = \begin{pmatrix} \mathbf{F}_x^2 * \mathbf{G}\mathbf{W}_\sigma & (\mathbf{F}_x \mathbf{F}_y) * \mathbf{G}\mathbf{W}_\sigma \\ (\mathbf{F}_y \mathbf{F}_x) * \mathbf{G}\mathbf{W}_\sigma & \mathbf{F}_y^2 * \mathbf{G}\mathbf{W}_\sigma \end{pmatrix} = \begin{pmatrix} A & C \\ C & B \end{pmatrix} \quad (\text{A.15})$$

Basierend auf der Determinante $\text{Det}(\mathbf{H}) = \alpha\beta = AB - C^2$ und der Spur (*Trace*) $\text{Tr}(\mathbf{H}) = \alpha + \beta = A + B$ wird in [Harris u. Stephens 1988] folgende Metrik zur Klassifikation in Eckpunkt oder Kante bzw. Fläche vorgeschlagen (Gl. A.16). Durch diese Herangehensweise wird die aufwendige Berechnung der Eigenwerte umgangen.

$$R_{\text{cornerness}} = \text{Det}(\mathbf{H}) - k\text{Tr}(\mathbf{H})^2 \quad (\text{A.16})$$

$$= \alpha\beta - k(\alpha + \beta)^2 = AB - C^2 - k(A + B)^2 \quad (\text{A.17})$$

Der Parameter $k \in (0 \dots 1)$ dient zur Festlegung der Klassengrenze zwischen Kante und Ecke. Wird k sehr klein gewählt, so werden auch Kanten als Ecken erkannt. Ist die Metrik $R_{\text{cornerness}}$ größer als Null, so handelt es sich um eine Ecke und nicht um eine Kante. Schwache Ecken werden ausgeschlossen, indem die Spur eine Schwelle T überschreiten muss (Gl. A.18).

$$R_{\text{cornerness}} > 0, \quad \text{Tr}(\mathbf{M}) = A + B > T \quad (\text{A.18})$$

Der Harris Detektor eignet sich zum Bestimmen von Ecken und auch Kanten im Bild. Dabei ist dieser invariant gegenüber Helligkeitsänderungen und gegen Rotation als auch gegen Verschiebung. Skalierung und Kontraständerungen beeinflussen jedoch die Verhaltensweise des Detektors.

SIFT – *Scale Invariant Feature Transform*

Das SIFT-Verfahren [Lowe 2004] besteht aus drei Komponenten (Detektion, Extraktion und Zuordnung), wobei der wesentliche Kern des Verfahrens die Extraktion darstellt. Im Folgenden soll der Ablauf kurz dargelegt werden.

Detektion (Detection)

Im ersten Schritt der Detektion wird vom Bild eine Repräsentation im Skalenraum (*Gaussian scale space*) erzeugt. Dies wird durchgeführt, um die Unabhängigkeit gegenüber Größenänderungen zu erreichen. Dazu wird das Bild mit einem Gaußkern $\mathbf{GW}_\sigma(x, y)$ unterschiedlicher Größen gefaltet. Die recheneffiziente Umsetzung (Abb. A.7) wendet eine wiederholte Filterung mit einem festen Kern $\mathbf{GW}_{k\sigma_1}(x, y)$ an. Zusätzlich wird je Verdoppelung (Oktave) der Größenordnung σ eine Unterabtastung durchgeführt. Eine Anzahl von $s = 3$ Skalen je Oktave zeigt in Vergleich zu anderen Werten s die höchste Leistungsfähigkeit [Lowe 2004] der Detektion ($k = 2^{1/s}$).

Ein Repräsentant (*feature point*) wird bestimmt, indem die Differenzen der Bilder der benachbarten Skalen gebildet werden. Punkte, die innerhalb ihrer 26er Nachbarschaft (Abb. A.7) ein Maximum bilden, sind potenzielle Merkmalspunkte, da es starke Gradienten in deren Umgebung gibt.

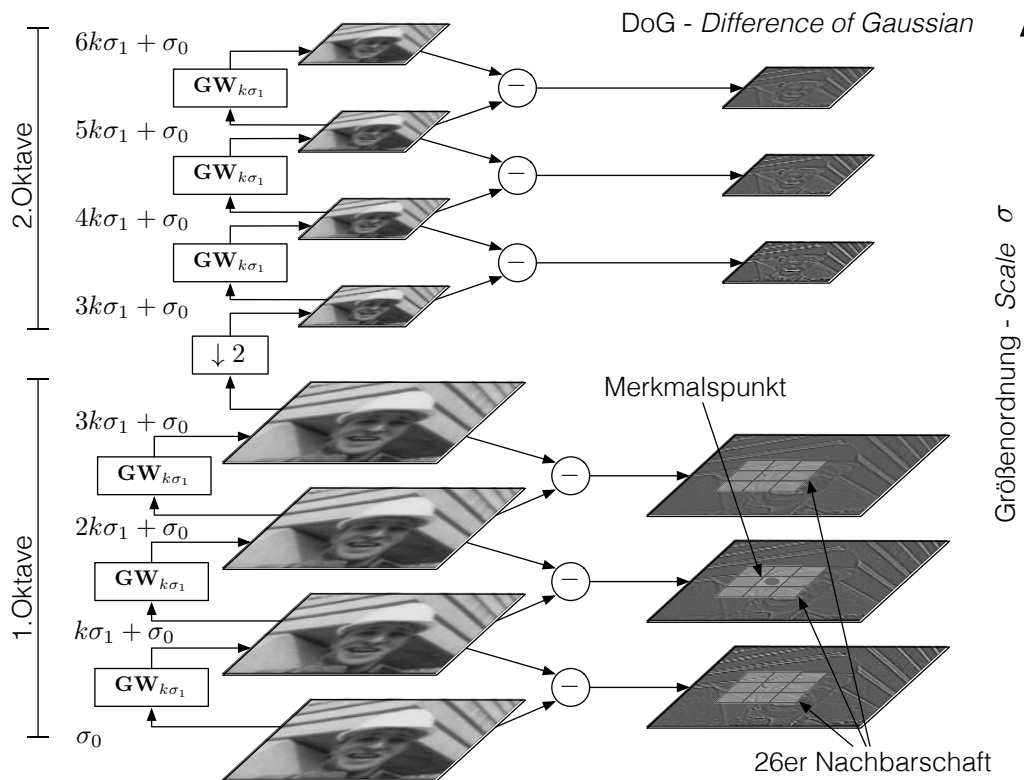


Abbildung A.7.: Recheneffiziente Umsetzung des Merkmalsdetektors auf Basis der *Difference of Gaussian* im SIFT-Algorithmus [Lowe 2004]

In weiteren Schritten wird die Sub-Pixel genaue Position bestimmt. Weiterhin werden Repräsentanten auf Basis der Hesse-Matrix aussortiert, welche ausschließlich eine Kante repräsentieren (starke Gradienten nur in einer Richtung). Hierbei wird eine ähnliche Metrik (*cornerness*) wie beim Harris-Detektor eingesetzt, jedoch die Hesse-Matrix direkt auf die Grauwerte des Bildes angewendet.

Abschließend wird die Ausrichtung jedes Repräsentanten durch Auswertung der dominanten Gradienten bestimmt. Gibt es mehrere dominante Ausrichtungen, so wird dies durch zwei Merkmalspunkte mit unterschiedlicher Ausrichtung dargestellt.

Jeder detektierte Repräsentant ist durch Position, Größenordnung (*scale*) und Ausrichtung beschrieben. Die Extraktion des Merkmalsvektors wird relativ zu diesen Eigenschaften durchgeführt, um Rotations-, Verschiebungs- und Skalierungsinvarianz zu gewährleisten.

Extraktion (Feature Extraction)

Der Merkmalsvektor enthält ausschließlich Eigenschaften der Gradienten in der Umgebung des Merkmalspunktes. Dazu werden diese in der Umgebung relativ zur Ausrichtung des Merkmalspunktes bestimmt und mit einer Gauß-Glocke gewichtet, um weit entfernte Gradienten weniger stark zu berücksichtigen. Die Umgebung wird weiterhin in 4×4 Bereiche unterteilt, wobei in jedem Bereich das Histogramm der Gradienten bestimmt wird. Gradienten nahe an Grenzen zwischen Bereichen werden geringer gewichtet, um die Einflüsse eines Wechsels zwischen Bereichen zu verringern. Jedes Histogramm wird in 8 Wertebereiche (*bin*) geteilt, wobei die jeweilige Häufigkeit ein Element des Merkmalsvektors ist. Im dargestellten Beispiel entsteht so ein Merkmalsvektor der Länge $4 \times 4 \times 8 = 128$. Es hat sich gezeigt [Lowe 2004], dass die Einteilung der Nachbarschaft in 4×4 Bereiche die höchste Leistungsfähigkeit liefert.

Zuordnung (Matching)

Die Zuordnung (*Matching*) der Merkmalspunkte zueinander wird anhand der geringsten euklidischen Distanz (*nearest neighbor*) zwischen den Merkmalsvektoren durchgeführt. Dabei werden ausschließlich Paarungen berücksichtigt, bei denen der zweit dichteste Merkmalsvektor eine wesentlich höhere Distanz als der dichteste Vektor aufweist. Dies führt zu einer robusteren Zuordnung.

Eigenschaften

Das SIFT-Verfahren ist sehr robust gegen diverse Störungen wie zum Beispiel Rauschen. Weiterhin ist es invariant gegenüber Rotation, Verschiebung und Skalierung. Helligkeits- oder Kontraständerungen sowie auch nichtlineare Pixeloperationen haben nur geringen Einfluss auf die Leistungsfähigkeit, da dieses Verfahren ausschließlich die Gradienten des Bildes verwendet.

B

Anhang B

Videosequenzen

In diesem Abschnitt werden kurz die in dieser Arbeit verwendeten Testsequenzen mit den zugehörigen technischen Merkmalen aufgeführt. Weiterhin wird eine kurze Einschätzung der für die Videocodierung wesentlichen Eigenschaften vorgenommen.

Bei Sequenzen mit mehreren Bildraten wurden die geringeren Bildwiederholraten erzeugt, indem die ungeraden Bilder aus den hochratigen Sequenzen entfernt wurden. Eine zeitliche Filterung wurde nicht durchgeführt.

B.1. QCIF



Bezeichnung: Akiyo

Bildgröße: 176×144 Pixel, QCIF

Farbformat: YCbCr 4:0:0

Quantisierung: 8 Bit

Bildrate: 30 bzw. 15 Bilder/s

Rohdatenrate: 6.08 bzw. 3.04 MBit/s

Anzahl der Bilder: 300 bzw. 150 Bilder

Charakteristik: statischer Hintergrund, im Wesentlichen nur Bewegung des Kopfes der Nachrichtensprecherin



Bezeichnung: **Coastguard**
Bildgröße: 176×144 Pixel, QCIF
Farbformat: YCbCr 4:0:0
Quantisierung: 8 Bit
Bildrate: 30 bzw. 15 Bilder/s
Rohdatenrate: 6.08 bzw. 3.04 MBit/s
Anzahl der Bilder: 300 bzw. 150 Bilder

Charakteristik: lineare Bewegung des Hintergrundes, Verdeckung und Aufdeckung im Vordergrund, Wasseroberfläche rauschähnlich



Bezeichnung: **Foreman**
Bildgröße: 176×144 Pixel, QCIF
Farbformat: YCbCr 4:0:0
Quantisierung: 8 Bit
Bildrate: 30 bzw. 15 Bilder/s
Rohdatenrate: 6.08 bzw. 3.04 MBit/s
Anzahl der Bilder: 300 bzw. 150 Bilder

Charakteristik: teilweise schnelle und nichtlineare Bewegung des Hintergrundes und Teile der Vordergrundobjekte



Bezeichnung: **Hall**
Bildgröße: 176×144 Pixel, QCIF
Farbformat: YCbCr 4:0:0
Quantisierung: 8 Bit
Bildrate: 30 bzw. 15 Bilder/s
Rohdatenrate: 6.08 bzw. 3.04 MBit/s
Anzahl der Bilder: 300 bzw. 150 Bilder

Charakteristik: statische Szene, nur Personenbewegungen



Bezeichnung: **Mother and Daughter**
Bildgröße: 176 × 144 Pixel, QCIF
Farbformat: YCbCr 4:0:0
Quantisierung: 8 Bit
Bildrate: 30 bzw. 15 Bilder/s
Rohdatenrate: 6.08 bzw. 3.04 MBit/s
Anzahl der Bilder: 300 bzw. 150 Bilder

Charakteristik: statischer Hintergrund, etwas stärkere Bewegung der Köpfe als bei Akiyo, Handbewegung



Bezeichnung: **Soccer**
Bildgröße: 176 × 144 Pixel, QCIF
Farbformat: YCbCr 4:0:0
Quantisierung: 8 Bit
Bildrate: 30 bzw. 15 Bilder/s
Rohdatenrate: 6.08 bzw. 3.04 MBit/s
Anzahl der Bilder: 300 bzw. 150 Bilder

Charakteristik: sehr starke nichtlineare Bewegung der Kamera, schnelle Bewegung des Vordergrundes, starke Verdeckungen



Bezeichnung: **Stefan**
Bildgröße: 176 × 144 Pixel, QCIF
Farbformat: YCbCr 4:0:0
Quantisierung: 8 Bit
Bildrate: 30 bzw. 15 Bilder/s
Rohdatenrate: 6.08 bzw. 3.04 MBit/s
Anzahl der Bilder: 300 bzw. 150 Bilder

Charakteristik: stark strukturierter statischer Hintergrund, schnelle Bewegung des Spielers, schnelle Kameraschwenks

B.2. CIF



Bezeichnung: Bus
Bildgröße: 352×288 Pixel, CIF
Farbformat: YCbCr 4:0:0
Quantisierung: 8 Bit
Bildrate: 30 Bilder/s
Rohdatenrate: 24.33 MBit/s
Anzahl der Bilder: 150 Bilder

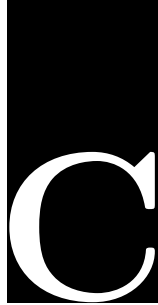
Charakteristik: starke lineare Bewegung von Vorder- und Hintergrund, starke Verdeckungen, periodische Bildelemente

B.3. SIF (NTSC)



Bezeichnung: SFlowG
Bildgröße: 352×240 Pixel, SIF
Farbformat: YCbCr 4:0:0
Quantisierung: 8 Bit
Bildrate: 30 Bilder/s
Rohdatenrate: 20.28 MBit/s
Anzahl der Bilder: 150 Bilder

Charakteristik: lineare Bewegung von Vorder- und Hintergrund, starke Verdeckungen, starke Textur bestimmter Bildbereiche



C.1. Mittlere Qualität der Seiteninformation

Die folgenden Tabellen C.2 bis C.9 stellen die mittlere Qualität der Seiteninformation für die verwendeten Testsequenzen dar. Dabei werden unterschiedliche Quantisierungsparameter der Key-Frames berücksichtigt. Die besten als auch die schlechtesten Ergebnisse sind **fett** beziehungsweise unterstrichen dargestellt.

Als Referenz werden die aus der Literatur bekannten Verfahren BiMESS [Ascenso u. a. 2005], MCTI [Ascenso u. Pereira 2008] und PBTI [Sofke u. a. 2009] verwendet.

Tabelle C.1 gibt die verwendeten Parameter der Bewegungsschätzung wieder. Alle Verfahren verwenden eine Halb-Pixel genaue Bewegungsschätzung.

	max. Suchweite		Matchingwindow- bzw. Blockgröße
	QCIF	CIF	
BiMESS [Ascenso u. a. 2005]	16 px	16 px	$8 \times 8 \text{ px}^2$
MCTI [Ascenso u. Pereira 2008]	32 px	32 px	$16 \times 16 \text{ px}^2$ bis $4 \times 4 \text{ px}^2$
PBTI [Sofke u. a. 2009]	10 px	15 px	$21 \times 21 \text{ px}^2$, $\sigma = 5 \text{ px}$
basicPBTI, GpTIE, APBTI(2)	10 px	15 px	$21 \times 21 \text{ px}^2$, $\sigma = 5 \text{ px}$
GTIE	10 px	16 px	$8 \times 8 \text{ px}^2$
MPBTI	6 px	9 px	$21 \times 21 \text{ px}^2$, $\sigma = 5 \text{ px}$
FPBTI	10 px	15 px	$15 \times 15 \text{ px}^2$, $\sigma = 5 \text{ px}$

Tabelle C.1.: Parameter für die Bewegungssuche

	BiMESS	MCTI	PBTI	basicPBTI	GTIE	GpTIE	MPBTI	APBTI	APBTI2	FPBTI	Key-F.
QCIF 30 fps											
Akiyo	49.2 dB	48.8 dB	50.3 dB	49.5 dB	49.0 dB	50.3 dB	49.5 dB	48.8 dB	49.4 dB	49.4 dB	∞ dB
Coastguard	34.6 dB	38.7 dB	39.1 dB	38.8 dB	37.7 dB	38.8 dB	38.8 dB	38.6 dB	38.9 dB	39.1 dB	∞ dB
Foreman	33.4 dB	35.6 dB	36.8 dB	35.9 dB	35.2 dB	36.5 dB	36.3 dB	35.3 dB	36.1 dB	35.9 dB	∞ dB
Hall	40.0 dB	40.8 dB	40.9 dB	40.9 dB	40.3 dB	40.8 dB	40.9 dB	40.6 dB	40.9 dB	41.1 dB	∞ dB
Mother & D.	41.5 dB	45.4 dB	46.6 dB	46.2 dB	45.1 dB	46.5 dB	46.2 dB	46.2 dB	46.1 dB	46.2 dB	∞ dB
Soccer	26.0 dB	28.1 dB	26.9 dB	26.8 dB	27.4 dB	28.3 dB	28.3 dB	26.5 dB	28.0 dB	26.7 dB	∞ dB
Stefan	26.3 dB	28.7 dB	27.3 dB	27.2 dB	27.8 dB	28.6 dB	28.9 dB	27.0 dB	28.8 dB	27.4 dB	∞ dB
CIF 30 fps											
Bus	26.0 dB	25.5 dB	25.7 dB	24.2 dB	25.3 dB	25.8 dB	27.0 dB	27.3 dB	26.9 dB	26.6 dB	∞ dB
SIF 30 fps											
SFlowG	27.1 dB	28.7 dB	29.3 dB	28.9 dB	27.6 dB	29.2 dB	29.0 dB	28.5 dB	28.8 dB	29.0 dB	∞ dB
QCIF 15 fps											
Akiyo	43.8 dB	44.0 dB	45.0 dB	44.5 dB	44.0 dB	45.0 dB	44.5 dB	43.9 dB	44.4 dB	44.4 dB	∞ dB
Coastguard	29.8 dB	32.8 dB	32.6 dB	32.7 dB	31.5 dB	32.7 dB	32.8 dB	32.4 dB	32.9 dB	32.9 dB	∞ dB
Foreman	28.4 dB	29.8 dB	30.0 dB	29.6 dB	29.5 dB	30.0 dB	30.2 dB	28.8 dB	30.0 dB	29.6 dB	∞ dB
Hall	36.6 dB	37.8 dB	37.6 dB	37.6 dB	36.8 dB	37.5 dB	37.7 dB	36.8 dB	37.5 dB	37.8 dB	∞ dB
Mother & D.	38.3 dB	40.6 dB	41.4 dB	41.0 dB	40.3 dB	41.3 dB	41.1 dB	40.9 dB	41.0 dB	41.1 dB	∞ dB
Soccer	21.6 dB	22.5 dB	21.4 dB	21.4 dB	22.4 dB	22.8 dB	22.9 dB	20.8 dB	22.4 dB	21.4 dB	∞ dB
Stefan	22.4 dB	23.6 dB	22.7 dB	22.7 dB	23.4 dB	23.6 dB	24.2 dB	22.1 dB	23.9 dB	22.8 dB	∞ dB
CIF 15 fps											
Bus	20.9 dB	23.2 dB	20.3 dB	19.8 dB	20.7 dB	21.0 dB	24.7 dB	20.5 dB	24.2 dB	20.8 dB	∞ dB
SIF 15 fps											
SFlowG	23.1 dB	24.3 dB	24.3 dB	22.3 dB	23.9 dB	24.6 dB	24.5 dB	23.9 dB	24.3 dB	24.2 dB	∞ dB

Tabelle C.2.: Mittlere Qualität (PSNR-Y) der Seiteninformation bei verlustfreien Key-Frames

	BiMESS	MCTI	PBTI	basicPBTI	GTIE	GpTIE	MPBTI	APBTI	APBTI2	FPBTI	Key-F.
QCIF 30 fps											
Akiyo	<u>41.7</u> dB	43.7 dB	44.3 dB	44.0 dB	43.5 dB	44.3 dB	44.0 dB	43.7 dB	43.9 dB	44.0 dB	44.9 dB
Coastguard	34.3 dB	37.5 dB	37.8 dB	37.7 dB	36.7 dB	37.7 dB	37.7 dB	37.5 dB	37.7 dB	37.8 dB	42.4 dB
Foreman	31.9 dB	34.8 dB	35.6 dB	34.9 dB	34.3 dB	35.4 dB	35.4 dB	34.5 dB	35.2 dB	34.9 dB	43.1 dB
Hall	<u>36.6</u> dB	39.8 dB	39.9 dB	39.9 dB	39.3 dB	39.8 dB	39.9 dB	39.7 dB	39.9 dB	40.0 dB	43.7 dB
Mother & D.	<u>36.3</u> dB	42.3 dB	43.0 dB	42.9 dB	41.8 dB	42.9 dB	42.8 dB	42.9 dB	42.8 dB	42.9 dB	44.3 dB
Soccer	25.8 dB	28.0 dB	26.8 dB	26.7 dB	27.4 dB	28.2 dB	28.2 dB	26.4 dB	27.9 dB	26.6 dB	42.8 dB
Stefan	<u>25.7</u> dB	28.5 dB	27.2 dB	27.1 dB	27.7 dB	28.5 dB	28.8 dB	26.9 dB	28.7 dB	27.3 dB	43.4 dB
CIF 30 fps											
Bus	25.7 dB	25.4 dB	25.7 dB	<u>24.1</u> dB	25.2 dB	25.8 dB	27.0 dB	27.2 dB	26.8 dB	26.6 dB	42.7 dB
SIF 30 fps											
SFlowG	<u>26.7</u> dB	28.6 dB	29.1 dB	28.8 dB	27.4 dB	29.0 dB	28.8 dB	28.4 dB	28.7 dB	28.8 dB	42.9 dB
QCIF 15 fps											
Akiyo	39.8 dB	41.5 dB	42.0 dB	41.8 dB	41.2 dB	42.0 dB	41.8 dB	41.4 dB	41.8 dB	41.8 dB	44.9 dB
Coastguard	29.6 dB	32.6 dB	32.4 dB	32.4 dB	31.3 dB	32.5 dB	32.6 dB	32.2 dB	32.6 dB	32.6 dB	42.4 dB
Foreman	<u>27.9</u> dB	29.6 dB	29.8 dB	29.4 dB	29.3 dB	29.7 dB	29.9 dB	28.6 dB	29.8 dB	29.4 dB	43.1 dB
Hall	<u>34.5</u> dB	37.2 dB	37.1 dB	37.1 dB	36.3 dB	37.0 dB	37.1 dB	36.4 dB	37.0 dB	37.3 dB	43.7 dB
Mother & D.	34.9 dB	39.3 dB	39.8 dB	39.6 dB	38.7 dB	39.7 dB	39.6 dB	39.6 dB	39.6 dB	39.6 dB	44.3 dB
Soccer	21.7 dB	22.5 dB	21.4 dB	21.4 dB	22.4 dB	22.8 dB	22.9 dB	<u>20.8</u> dB	22.4 dB	21.4 dB	42.8 dB
Stefan	22.3 dB	23.6 dB	22.7 dB	22.7 dB	23.4 dB	23.6 dB	24.2 dB	<u>22.1</u> dB	23.9 dB	22.8 dB	43.4 dB
CIF 15 fps											
Bus	20.9 dB	23.2 dB	20.4 dB	<u>19.8</u> dB	20.7 dB	21.0 dB	24.7 dB	20.5 dB	24.1 dB	20.9 dB	42.7 dB
SIF 15 fps											
SFlowG	23.1 dB	24.2 dB	24.3 dB	<u>22.3</u> dB	23.8 dB	24.6 dB	24.4 dB	23.9 dB	24.3 dB	24.2 dB	42.9 dB

Tabelle C.3.: Mittlere Qualität (PSNR-Y) der Seiteninformation, Key-Frame: H.264intra, $QP = 20$

	BiMESS	MCTI	PBTI	basicPBTI	GTIE	G _p TIE	MPBTI	APBTI	APBTI2	FPBTI	Key-F.
QCIF 30 fps											
Akiyo	<u>37.7</u> dB	41.1 dB	41.4 dB	41.2 dB	40.9 dB	41.4 dB	41.2 dB	41.1 dB	41.2 dB	41.2 dB	41.3 dB
Coastguard	<u>33.4</u> dB	36.0 dB	36.3 dB	36.2 dB	35.3 dB	36.2 dB	36.2 dB	36.1 dB	36.3 dB	36.3 dB	38.2 dB
Foreman	30.2 dB	33.9 dB	34.5 dB	34.0 dB	33.4 dB	34.4 dB	34.5 dB	33.7 dB	34.3 dB	34.1 dB	39.2 dB
Hall	<u>34.5</u> dB	38.5 dB	38.6 dB	38.6 dB	38.0 dB	38.6 dB	38.6 dB	38.5 dB	38.6 dB	38.7 dB	40.4 dB
Mother & D.	<u>34.8</u> dB	39.9 dB	40.4 dB	40.3 dB	39.4 dB	40.3 dB	40.3 dB	40.3 dB	40.3 dB	40.3 dB	40.6 dB
Soccer	<u>25.4</u> dB	27.8 dB	26.7 dB	26.5 dB	27.3 dB	28.1 dB	28.0 dB	26.2 dB	27.7 dB	26.5 dB	39.0 dB
Stefan	<u>25.3</u> dB	28.3 dB	27.0 dB	26.9 dB	27.5 dB	28.3 dB	28.6 dB	26.8 dB	28.5 dB	27.1 dB	39.0 dB
CIF 30 fps											
Bus	25.4 dB	25.3 dB	25.6 dB	<u>24.1</u> dB	25.1 dB	25.8 dB	26.8 dB	27.0 dB	26.7 dB	26.4 dB	38.5 dB
SIF 30 fps											
SFlowG	<u>26.4</u> dB	28.4 dB	28.9 dB	28.5 dB	27.2 dB	28.8 dB	28.6 dB	28.2 dB	28.4 dB	28.6 dB	38.2 dB
QCIF 15 fps											
Akiyo	<u>37.0</u> dB	39.7 dB	40.1 dB	40.0 dB	39.5 dB	40.1 dB	40.0 dB	39.7 dB	39.9 dB	39.9 dB	41.3 dB
Coastguard	<u>29.2</u> dB	32.1 dB	32.0 dB	32.0 dB	31.0 dB	32.1 dB	32.2 dB	31.9 dB	32.2 dB	32.1 dB	38.2 dB
Foreman	<u>26.9</u> dB	29.4 dB	29.4 dB	29.1 dB	29.0 dB	29.4 dB	29.7 dB	28.4 dB	29.5 dB	29.1 dB	39.2 dB
Hall	<u>33.1</u> dB	36.3 dB	36.3 dB	36.4 dB	35.6 dB	36.3 dB	36.4 dB	35.8 dB	36.3 dB	36.5 dB	40.4 dB
Mother & D.	<u>33.9</u> dB	38.0 dB	38.3 dB	38.2 dB	37.4 dB	38.2 dB	38.2 dB	38.2 dB	38.2 dB	38.2 dB	40.6 dB
Soccer	21.5 dB	22.4 dB	21.4 dB	21.4 dB	22.4 dB	22.8 dB	22.8 dB	<u>20.8</u> dB	22.4 dB	21.3 dB	39.0 dB
Stefan	<u>22.1</u> dB	23.5 dB	22.6 dB	22.7 dB	23.3 dB	23.6 dB	24.1 dB	<u>22.1</u> dB	23.9 dB	22.7 dB	38.9 dB
CIF 15 fps											
Bus	20.8 dB	23.1 dB	20.4 dB	<u>19.8</u> dB	20.7 dB	21.0 dB	24.6 dB	20.5 dB	24.0 dB	20.8 dB	38.5 dB
SIF 15 fps											
SFlowG	23.0 dB	24.1 dB	24.2 dB	<u>22.3</u> dB	23.8 dB	24.6 dB	24.4 dB	23.9 dB	24.2 dB	24.1 dB	38.2 dB

Tabelle C.4.: Mittlere Qualität (PSNR-Y) der Seiteninformation, Key-Frame: H.264intra, $QP = 25$

	BiMESS	MCTI	PBTI	basicPBTI	GTIE	GpTIE	MPBTI	APBTI	APBTI2	FPBTI	Key-F.
QCIF 30 fps											
Akiyo	<u>34.8</u> dB	37.8 dB	37.9 dB	37.9 dB	37.6 dB	37.9 dB	37.9 dB	37.8 dB	37.9 dB	37.9 dB	37.6 dB
Coastguard	31.1 dB	33.5 dB	33.8 dB	33.7 dB	33.0 dB	33.7 dB	33.7 dB	33.6 dB	33.7 dB	33.7 dB	34.1 dB
Foreman	28.3 dB	32.4 dB	32.8 dB	32.5 dB	32.1 dB	32.7 dB	32.9 dB	32.3 dB	32.7 dB	32.6 dB	35.4 dB
Hall	<u>32.3</u> dB	36.1 dB	36.2 dB	36.2 dB	35.6 dB	36.1 dB	36.2 dB	36.1 dB	36.2 dB	36.2 dB	36.5 dB
Mother & D.	<u>33.1</u> dB	36.7 dB	37.1 dB	37.0 dB	36.3 dB	37.0 dB	37.0 dB	37.1 dB	37.0 dB	37.0 dB	36.9 dB
Soccer	24.7 dB	27.3 dB	26.4 dB	26.2 dB	26.9 dB	27.8 dB	27.6 dB	25.9 dB	27.3 dB	26.1 dB	35.4 dB
Stefan	<u>24.9</u> dB	27.7 dB	26.5 dB	26.4 dB	27.0 dB	27.7 dB	28.0 dB	26.4 dB	27.9 dB	26.6 dB	34.2 dB
CIF 30 fps											
Bus	24.7 dB	25.1 dB	25.4 dB	<u>23.9</u> dB	24.7 dB	25.5 dB	26.5 dB	26.7 dB	26.3 dB	26.1 dB	34.2 dB
SIF 30 fps											
SFlowG	<u>25.8</u> dB	27.7 dB	28.2 dB	28.0 dB	26.7 dB	28.2 dB	28.0 dB	27.7 dB	27.9 dB	28.0 dB	33.2 dB
QCIF 15 fps											
Akiyo	34.1 dB	37.1 dB	37.3 dB	37.3 dB	36.9 dB	37.3 dB	37.3 dB	37.2 dB	37.3 dB	37.3 dB	37.6 dB
Coastguard	28.3 dB	31.0 dB	31.0 dB	31.0 dB	30.2 dB	31.1 dB	31.1 dB	30.9 dB	31.1 dB	31.1 dB	34.1 dB
Foreman	<u>25.8</u> dB	28.8 dB	28.8 dB	28.6 dB	28.4 dB	28.8 dB	29.1 dB	28.0 dB	28.9 dB	28.5 dB	35.4 dB
Hall	31.4 dB	34.7 dB	34.7 dB	34.7 dB	34.0 dB	34.7 dB	34.7 dB	34.3 dB	34.6 dB	34.8 dB	36.5 dB
Mother & D.	32.4 dB	35.6 dB	36.0 dB	36.0 dB	35.1 dB	35.9 dB	36.0 dB	35.9 dB	35.9 dB	36.0 dB	36.9 dB
Soccer	21.2 dB	22.2 dB	21.4 dB	21.3 dB	22.3 dB	22.7 dB	22.8 dB	<u>20.8</u> dB	22.3 dB	21.3 dB	35.4 dB
Stefan	<u>21.9</u> dB	23.4 dB	22.5 dB	22.5 dB	23.2 dB	23.5 dB	24.0 dB	22.1 dB	23.7 dB	22.6 dB	34.2 dB
CIF 15 fps											
Bus	20.6 dB	23.0 dB	20.4 dB	<u>19.8</u> dB	20.7 dB	21.0 dB	24.4 dB	20.5 dB	23.9 dB	20.8 dB	34.2 dB
SIF 15 fps											
SFlowG	22.8 dB	23.9 dB	24.0 dB	<u>22.2</u> dB	23.5 dB	24.3 dB	24.2 dB	23.7 dB	24.0 dB	23.9 dB	33.2 dB

Tabelle C.5.: Mittlere Qualität (PSNR-Y) der Seiteninformation, Key-Frame: H.264intra, $QP = 30$

	BiMESS	MCTI	PBTI	basicPBTI	GTIE	GpTIE	MPBTI	APBTI	APBTI2	FPBTI	Key-F.
QCIF 30 fps											
Akiyo	<u>32.2</u> dB	34.2 dB	34.3 dB	34.3 dB	34.1 dB	34.3 dB	34.3 dB	34.2 dB	34.3 dB	34.3 dB	33.8 dB
Coastguard	<u>28.1</u> dB	30.7 dB	31.0 dB	30.9 dB	30.4 dB	30.9 dB	30.9 dB	30.9 dB	30.9 dB	30.9 dB	30.7 dB
Foreman	26.5 dB	30.4 dB	30.7 dB	30.6 dB	30.2 dB	30.6 dB	30.9 dB	30.4 dB	30.8 dB	30.6 dB	31.9 dB
Hall	<u>29.9</u> dB	32.9 dB	33.0 dB	33.0 dB	32.6 dB	33.0 dB	33.1 dB	33.0 dB	33.0 dB	33.1 dB	32.8 dB
Mother & D.	<u>31.9</u> dB	33.8 dB	34.0 dB	33.9 dB	33.5 dB	33.9 dB	34.0 dB	34.0 dB	33.9 dB	33.9 dB	33.7 dB
Soccer	<u>23.9</u> dB	26.6 dB	26.0 dB	25.8 dB	26.5 dB	27.3 dB	27.1 dB	25.4 dB	26.8 dB	25.6 dB	32.6 dB
Stefan	<u>24.1</u> dB	26.5 dB	25.4 dB	25.4 dB	26.0 dB	26.5 dB	26.9 dB	25.4 dB	26.8 dB	25.5 dB	29.8 dB
CIF 30 fps											
Bus	23.7 dB	24.6 dB	24.9 dB	<u>23.6</u> dB	24.1 dB	25.1 dB	25.8 dB	25.9 dB	25.6 dB	25.4 dB	30.5 dB
SIF 30 fps											
SFlowG	<u>24.5</u> dB	26.3 dB	26.7 dB	26.6 dB	25.5 dB	26.7 dB	26.6 dB	26.3 dB	26.5 dB	26.6 dB	28.6 dB
QCIF 15 fps											
Akiyo	<u>31.7</u> dB	33.9 dB	34.1 dB	34.0 dB	33.7 dB	34.1 dB	34.1 dB	34.0 dB	34.0 dB	34.0 dB	33.8 dB
Coastguard	<u>26.6</u> dB	29.3 dB	29.4 dB	29.4 dB	28.8 dB	29.4 dB	29.5 dB	29.3 dB	29.4 dB	29.4 dB	30.7 dB
Foreman	<u>24.8</u> dB	27.8 dB	27.8 dB	27.6 dB	27.5 dB	27.9 dB	28.2 dB	27.1 dB	28.0 dB	27.6 dB	31.9 dB
Hall	<u>29.2</u> dB	32.2 dB	32.3 dB	32.3 dB	31.6 dB	32.3 dB	32.3 dB	32.0 dB	32.2 dB	32.3 dB	32.8 dB
Mother & D.	<u>31.4</u> dB	33.3 dB	33.5 dB	33.5 dB	33.0 dB	33.4 dB	33.5 dB	33.5 dB	33.4 dB	33.4 dB	33.7 dB
Soccer	20.9 dB	22.0 dB	21.3 dB	21.2 dB	22.2 dB	22.6 dB	22.6 dB	<u>20.7</u> dB	22.2 dB	21.2 dB	32.6 dB
Stefan	<u>21.6</u> dB	23.0 dB	22.2 dB	22.3 dB	22.9 dB	23.1 dB	23.6 dB	21.9 dB	23.4 dB	22.3 dB	29.8 dB
CIF 15 fps											
Bus	20.3 dB	22.7 dB	20.4 dB	<u>19.8</u> dB	20.5 dB	21.0 dB	24.0 dB	20.4 dB	23.4 dB	20.8 dB	30.5 dB
SIF 15 fps											
SFlowG	22.2 dB	23.3 dB	23.5 dB	<u>21.9</u> dB	22.9 dB	23.7 dB	23.6 dB	23.2 dB	23.5 dB	23.4 dB	28.6 dB

Tabelle C.6.: Mittlere Qualität (PSNR-Y) der Seiteninformation, Key-Frame: H.264intra, $QP = 35$

	BiMESS	MCTI	PBTI	basicPBTI	GTIE	GpTIE	MPBTI	APBTI	APBTI2	FPBTI	Key-F.
QCIF 30 fps											
Akiyo	<u>29.6</u> dB	31.0 dB	31.1 dB	31.1 dB	30.8 dB	31.1 dB	31.1 dB	31.0 dB	31.0 dB	31.0 dB	30.7 dB
Coastguard	25.6 dB	27.9 dB	28.2 dB	28.2 dB	27.7 dB	28.1 dB	28.2 dB	28.1 dB	28.0 dB	28.1 dB	27.8 dB
Foreman	25.0 dB	27.8 dB	28.4 dB	28.3 dB	27.8 dB	28.3 dB	28.5 dB	28.2 dB	28.4 dB	28.3 dB	28.8 dB
Hall	<u>27.8</u> dB	29.6 dB	29.7 dB	29.7 dB	29.4 dB	29.7 dB	29.7 dB	29.7 dB	29.7 dB	29.7 dB	29.2 dB
Mother & D.	29.5 dB	30.9 dB	31.0 dB	31.0 dB	30.6 dB	31.0 dB	31.0 dB	31.0 dB	31.0 dB	30.9 dB	30.7 dB
Soccer	23.4 dB	25.7 dB	25.4 dB	25.2 dB	25.9 dB	26.6 dB	26.4 dB	24.8 dB	26.2 dB	25.1 dB	30.0 dB
Stefan	<u>22.9</u> dB	24.6 dB	23.7 dB	23.8 dB	24.2 dB	24.6 dB	24.9 dB	23.8 dB	24.8 dB	23.8 dB	26.0 dB
CIF 30 fps											
Bus	<u>22.2</u> dB	23.6 dB	24.1 dB	23.0 dB	23.0 dB	24.2 dB	24.7 dB	24.6 dB	24.5 dB	24.3 dB	27.3 dB
SIF 30 fps											
SFlowG	<u>22.3</u> dB	24.0 dB	24.4 dB	24.3 dB	23.3 dB	24.3 dB	24.4 dB	24.1 dB	24.3 dB	24.3 dB	24.6 dB
QCIF 15 fps											
Akiyo	29.4 dB	30.9 dB	31.0 dB	31.0 dB	30.7 dB	31.0 dB	31.0 dB	31.0 dB	31.0 dB	31.0 dB	30.7 dB
Coastguard	24.9 dB	27.3 dB	27.4 dB	27.4 dB	26.9 dB	27.4 dB	27.5 dB	27.3 dB	27.3 dB	27.3 dB	27.8 dB
Foreman	<u>23.8</u> dB	26.1 dB	26.4 dB	26.3 dB	26.1 dB	26.5 dB	26.7 dB	25.9 dB	26.7 dB	26.3 dB	28.8 dB
Hall	27.3 dB	29.2 dB	29.4 dB	29.4 dB	28.9 dB	29.4 dB	29.4 dB	29.2 dB	29.3 dB	29.4 dB	29.2 dB
Mother & D.	29.1 dB	30.6 dB	30.8 dB	30.8 dB	30.4 dB	30.8 dB	30.8 dB	30.8 dB	30.7 dB	30.7 dB	30.7 dB
Soccer	<u>20.6</u> dB	21.9 dB	21.2 dB	21.1 dB	21.9 dB	22.3 dB	22.3 dB	<u>20.6</u> dB	21.9 dB	21.0 dB	30.0 dB
Stefan	<u>21.0</u> dB	22.3 dB	21.6 dB	21.6 dB	22.1 dB	22.4 dB	22.8 dB	21.4 dB	22.6 dB	21.6 dB	26.0 dB
CIF 15 fps											
Bus	<u>19.8</u> dB	22.1 dB	20.3 dB	<u>19.9</u> dB	20.2 dB	20.8 dB	23.2 dB	20.3 dB	22.6 dB	20.6 dB	27.3 dB
SIF 15 fps											
SFlowG	<u>21.0</u> dB	22.1 dB	22.3 dB	21.2 dB	21.7 dB	22.5 dB	22.4 dB	22.1 dB	22.3 dB	22.2 dB	24.6 dB

Tabelle C.7.: Mittlere Qualität (PSNR-Y) der Seiteninformation, Key-Frame: H.264intra, $QP = 40$

	BiMESS	MCTI	PBTI	basicPBTI	GTIE	GpTIE	MPBTI	APBTI	APBTI2	FPBTI	Key-F.
QCIF 30 fps											
Akiyo	<u>27.0</u> dB	28.0 dB	28.1 dB	28.1 dB	27.8 dB	28.1 dB	28.1 dB	28.0 dB	28.1 dB	28.1 dB	27.8 dB
Coastguard	<u>23.7</u> dB	25.3 dB	25.5 dB	25.4 dB	25.0 dB	25.5 dB	25.5 dB	25.3 dB	25.3 dB	25.3 dB	25.1 dB
Foreman	23.5 dB	25.2 dB	25.8 dB	25.7 dB	25.4 dB	25.7 dB	25.8 dB	25.6 dB	25.7 dB	25.6 dB	25.9 dB
Hall	<u>25.1</u> dB	25.9 dB	26.1 dB	26.1 dB	25.7 dB	26.0 dB	26.1 dB	25.9 dB	26.0 dB	26.0 dB	25.6 dB
Mother & D.	<u>27.6</u> dB	28.2 dB	28.3 dB	28.2 dB	28.1 dB	28.3 dB	28.3 dB	28.2 dB	28.2 dB	28.2 dB	28.1 dB
Soccer	<u>22.7</u> dB	24.3 dB	24.5 dB	24.3 dB	24.9 dB	25.5 dB	25.2 dB	23.9 dB	25.1 dB	24.1 dB	27.5 dB
Stefan	<u>21.0</u> dB	22.1 dB	21.7 dB	21.7 dB	21.6 dB	22.1 dB	22.3 dB	21.6 dB	22.2 dB	21.6 dB	22.6 dB
CIF 30 fps											
Bus	<u>20.8</u> dB	22.2 dB	22.7 dB	22.0 dB	21.5 dB	22.8 dB	23.1 dB	22.8 dB	22.9 dB	22.7 dB	24.3 dB
SIF 30 fps											
SFlowG	<u>19.6</u> dB	21.0 dB	21.3 dB	21.3 dB	20.1 dB	21.3 dB	21.4 dB	21.1 dB	21.3 dB	21.2 dB	21.0 dB
QCIF 15 fps											
Akiyo	<u>26.8</u> dB	28.0 dB	28.1 dB	28.1 dB	27.7 dB	28.1 dB	28.1 dB	28.0 dB	28.1 dB	28.1 dB	27.8 dB
Coastguard	<u>23.3</u> dB	25.0 dB	25.1 dB	25.1 dB	24.7 dB	25.2 dB	25.2 dB	25.0 dB	25.0 dB	25.0 dB	25.1 dB
Foreman	<u>22.6</u> dB	24.2 dB	24.6 dB	24.5 dB	24.3 dB	24.7 dB	24.8 dB	24.2 dB	24.7 dB	24.5 dB	25.9 dB
Hall	<u>24.8</u> dB	25.7 dB	25.9 dB	25.9 dB	25.5 dB	25.9 dB	25.9 dB	25.8 dB	25.9 dB	25.9 dB	25.6 dB
Mother & D.	<u>27.5</u> dB	28.1 dB	28.2 dB	28.2 dB	28.0 dB	28.2 dB	28.2 dB	28.1 dB	28.1 dB	28.1 dB	28.1 dB
Soccer	<u>20.5</u> dB	21.4 dB	21.0 dB	20.9 dB	21.9 dB	22.2 dB	22.0 dB	<u>20.4</u> dB	21.7 dB	20.9 dB	27.5 dB
Stefan	<u>20.0</u> dB	21.0 dB	20.6 dB	20.6 dB	20.6 dB	21.0 dB	21.3 dB	20.4 dB	21.1 dB	20.5 dB	22.6 dB
CIF 15 fps											
Bus	<u>19.2</u> dB	21.2 dB	20.1 dB	19.8 dB	19.7 dB	20.6 dB	22.1 dB	19.9 dB	21.6 dB	20.3 dB	24.3 dB
SIF 15 fps											
SFlowG	<u>19.0</u> dB	20.1 dB	20.3 dB	19.8 dB	19.3 dB	20.4 dB	20.4 dB	20.1 dB	20.3 dB	20.2 dB	21.0 dB

Tabelle C.8.: Mittlere Qualität (PSNR-Y) der Seiteninformation, Key-Frame: H.264intra, $QP = 45$

	BiMESS	MCTI	PBTI	basicPBTI	GTIE	GpTIE	MPBTI	APBTI	APBTI2	FPBTI	Key-F.
QCIF 30 fps											
Akiyo	<u>24.8</u> dB	25.2 dB	25.2 dB	25.2 dB	25.1 dB	25.2 dB	25.2 dB	25.2 dB	25.2 dB	25.2 dB	25.1 dB
Coastguard	<u>22.1</u> dB	22.8 dB	23.0 dB	23.0 dB	22.7 dB	23.0 dB	23.0 dB	22.8 dB	22.9 dB	22.9 dB	22.8 dB
Foreman	<u>22.2</u> dB	22.6 dB	23.0 dB	22.9 dB	22.8 dB	22.9 dB	23.0 dB	22.8 dB	22.9 dB	22.8 dB	22.9 dB
Hall	<u>22.2</u> dB	22.6 dB	22.7 dB	22.7 dB	22.5 dB	22.7 dB	22.7 dB	22.6 dB	22.6 dB	22.6 dB	22.5 dB
Mother & D.	<u>25.5</u> dB	25.6 dB	25.7 dB	25.7 dB	25.6 dB	25.7 dB	25.7 dB	25.6 dB	25.6 dB	25.6 dB	25.6 dB
Soccer	<u>22.2</u> dB	23.1 dB	23.4 dB	23.2 dB	23.6 dB	24.0 dB	23.8 dB	22.8 dB	23.7 dB	23.1 dB	24.9 dB
Stefan	<u>19.3</u> dB	19.8 dB	19.8 dB	19.8 dB	19.6 dB	19.9 dB	20.0 dB	19.7 dB	19.8 dB	19.7 dB	20.1 dB
CIF 30 fps											
Bus	<u>19.5</u> dB	20.3 dB	20.9 dB	20.5 dB	19.9 dB	20.9 dB	20.9 dB	20.5 dB	20.7 dB	20.6 dB	21.5 dB
SIF 30 fps											
SFlowG	<u>17.0</u> dB	17.9 dB	18.3 dB	18.3 dB	<u>17.0</u> dB	18.3 dB	18.3 dB	17.8 dB	18.1 dB	17.9 dB	18.1 dB
QCIF 15 fps											
Akiyo	24.8 dB	25.2 dB	25.3 dB	25.2 dB	25.1 dB	25.3 dB	25.3 dB	25.2 dB	25.2 dB	25.2 dB	25.1 dB
Coastguard	22.0 dB	22.7 dB	22.9 dB	22.8 dB	22.6 dB	22.9 dB	22.9 dB	22.7 dB	22.8 dB	22.8 dB	22.8 dB
Foreman	<u>21.7</u> dB	22.1 dB	22.4 dB	22.3 dB	22.4 dB	22.5 dB	22.5 dB	22.1 dB	22.4 dB	22.3 dB	22.9 dB
Hall	<u>22.2</u> dB	22.5 dB	22.7 dB	22.6 dB	22.4 dB	22.6 dB	22.6 dB	22.5 dB	22.6 dB	22.6 dB	22.5 dB
Mother & D.	25.5 dB	25.6 dB	25.7 dB	25.7 dB	25.6 dB	25.7 dB	25.7 dB	25.6 dB	25.6 dB	25.6 dB	25.6 dB
Soccer	20.3 dB	21.0 dB	20.8 dB	20.7 dB	21.5 dB	21.8 dB	21.5 dB	<u>20.1</u> dB	21.3 dB	20.6 dB	24.9 dB
Stefan	<u>18.9</u> dB	19.4 dB	19.4 dB	19.4 dB	19.3 dB	19.4 dB	19.6 dB	19.2 dB	19.4 dB	19.3 dB	20.1 dB
CIF 15 fps											
Bus	<u>18.7</u> dB	19.8 dB	19.5 dB	19.4 dB	19.1 dB	19.9 dB	20.3 dB	19.1 dB	20.0 dB	19.6 dB	21.5 dB
SIF 15 fps											
SFlowG	<u>16.7</u> dB	17.6 dB	17.8 dB	17.7 dB	<u>16.8</u> dB	17.9 dB	17.9 dB	17.5 dB	17.8 dB	17.5 dB	18.1 dB

Tabelle C.9.: Mittlere Qualität (PSNR-Y) der Seiteninformation, Key-Frame: H.264intra, $QP = 50$

C.2. Qualität der Seiteninformation per Frame

Dieser Abschnitt zeigt die Qualität der Seiteninformation für jeden Frame des RD-Punktes 2. Die H.264intra Key-Frames wurden mit einen Quantisierungsparameter $QP = 35$ codiert.

Akiyo – QCIF

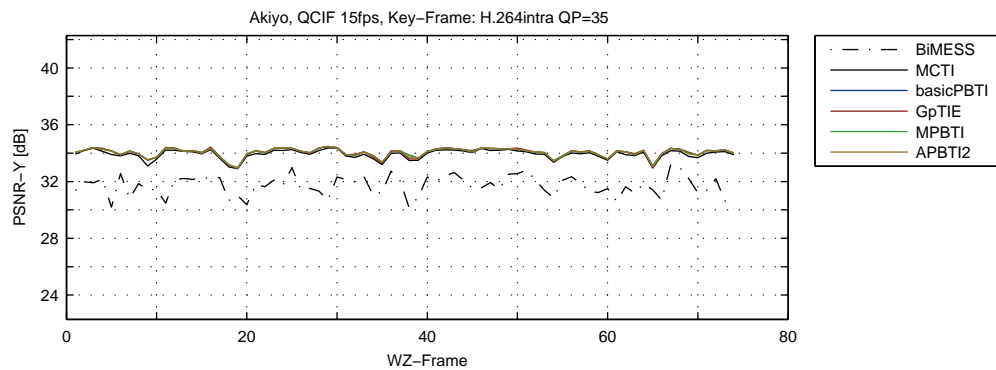


Abbildung C.1.: Qualität der Seiteninformation per WZ-Frame, Akiyo, QCIF, 15 fps, Key-Frames: H.264intra, $QP = 35$

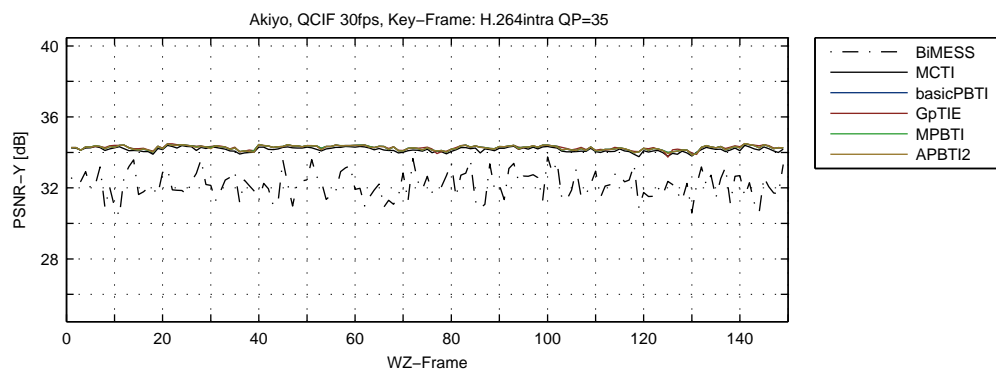


Abbildung C.2.: Qualität der Seiteninformation per WZ-Frame, Akiyo, QCIF, 30 fps, Key-Frames: H.264intra, $QP = 35$

Coastguard – QCIF

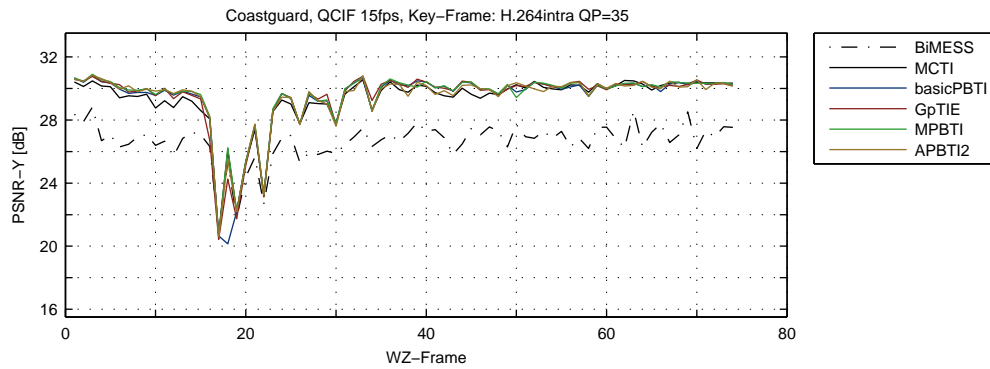


Abbildung C.3.: Qualität der Seiteninformation per WZ-Frame, Coastguard, QCIF, 15 fps, Key-Frames: H.264intra, $QP = 35$

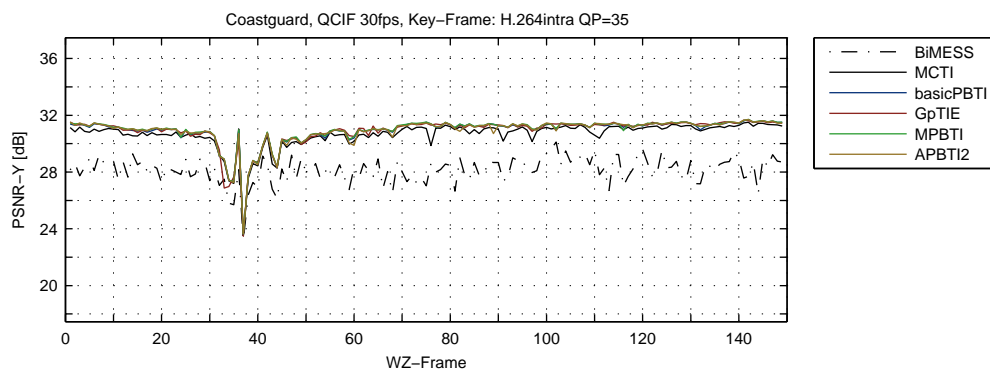


Abbildung C.4.: Qualität der Seiteninformation per WZ-Frame, Coastguard, QCIF, 30 fps, Key-Frames: H.264intra, $QP = 35$

Foreman – QCIF

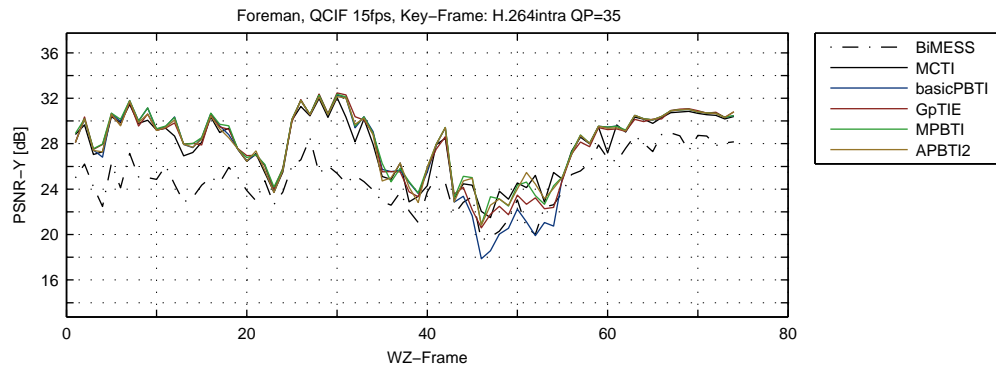


Abbildung C.5.: Qualität der Seiteninformation per WZ-Frame, Foreman, QCIF, 15 fps, Key-Frames: H.264intra, $QP = 35$

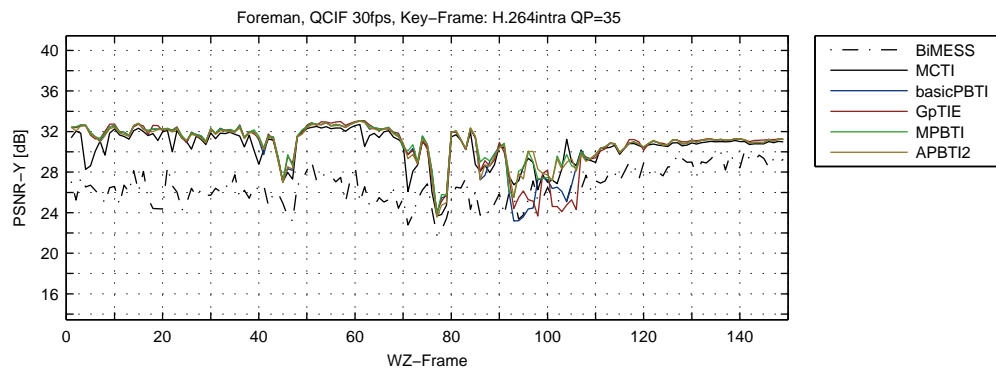


Abbildung C.6.: Qualität der Seiteninformation per WZ-Frame, Foreman, QCIF, 30 fps, Key-Frames: H.264intra, $QP = 35$

Ha11 – QCIF

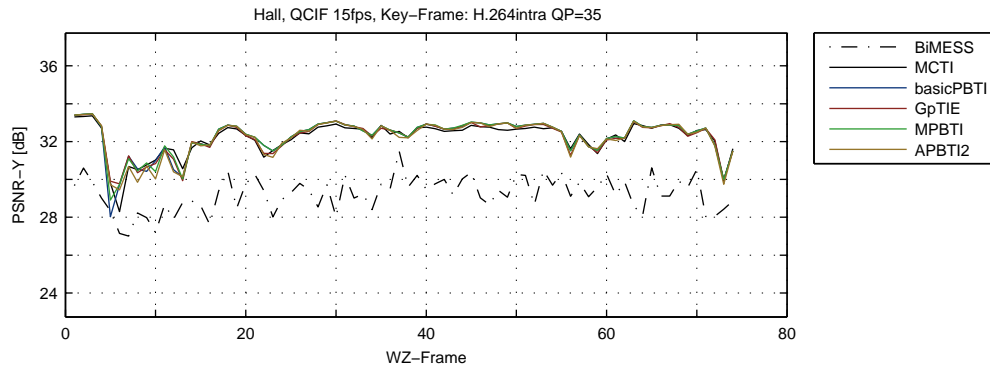


Abbildung C.7.: Qualität der Seiteninformation per WZ-Frame, Ha11, QCIF, 15 fps, Key-Frames: H.264intra, $QP = 35$

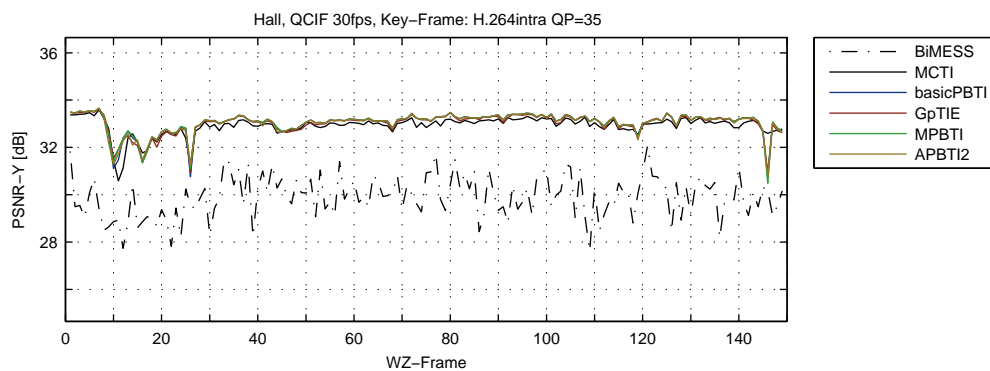


Abbildung C.8.: Qualität der Seiteninformation per WZ-Frame, Ha11, QCIF, 30 fps, Key-Frames: H.264intra, $QP = 35$

Mother and Daughter – QCIF

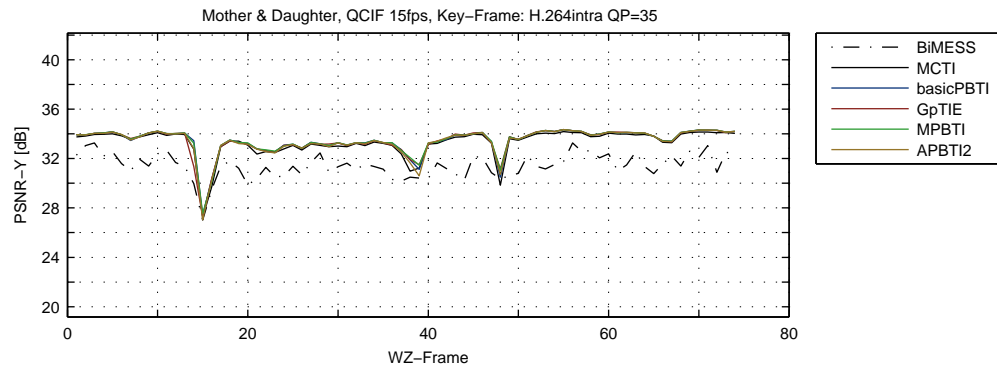


Abbildung C.9.: Qualität der Seiteninformation per WZ-Frame, Mother and Daughter, QCIF, 15 fps, Key-Frames: H.264intra, $QP = 35$

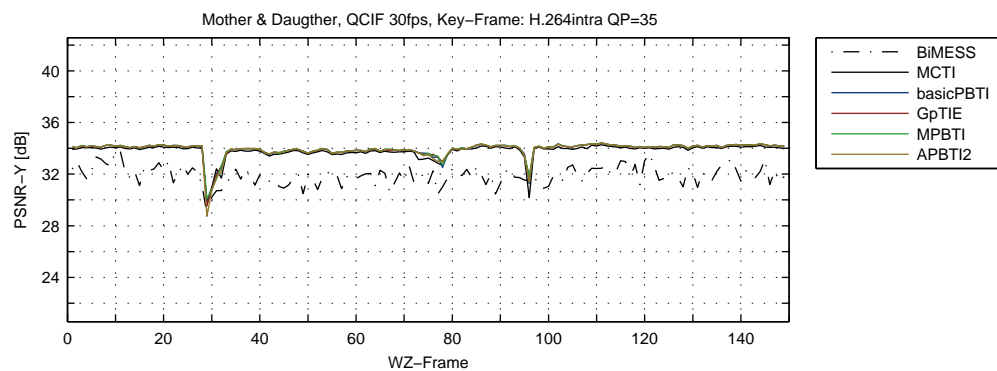


Abbildung C.10.: Qualität der Seiteninformation per WZ-Frame, Mother and Daughter, QCIF, 30 fps, Key-Frames: H.264intra, $QP = 35$

Soccer – QCIF

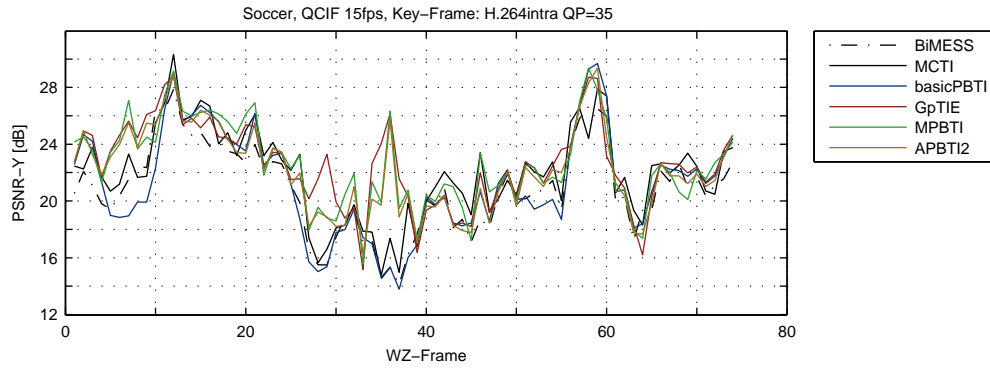


Abbildung C.11.: Qualität der Seiteninformation per WZ-Frame, Soccer, QCIF, 15 fps, Key-Frames: H.264intra, $QP = 35$

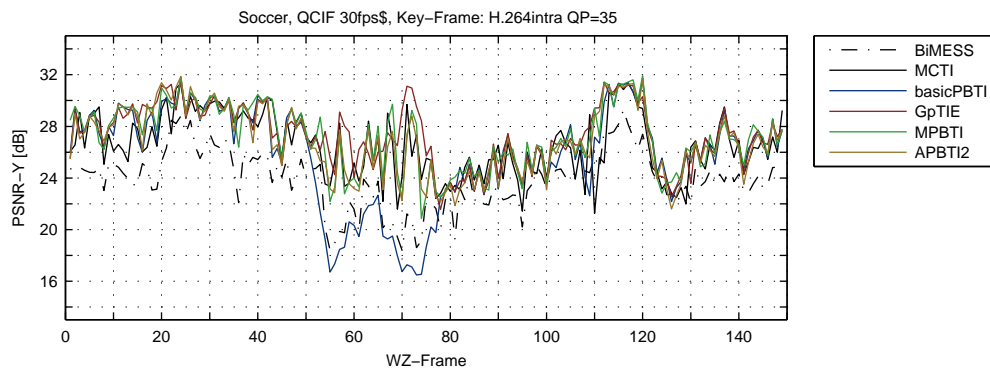


Abbildung C.12.: Qualität der Seiteninformation per WZ-Frame, Soccer, QCIF, 30 fps, Key-Frames: H.264intra, $QP = 35$

Stefan – QCIF

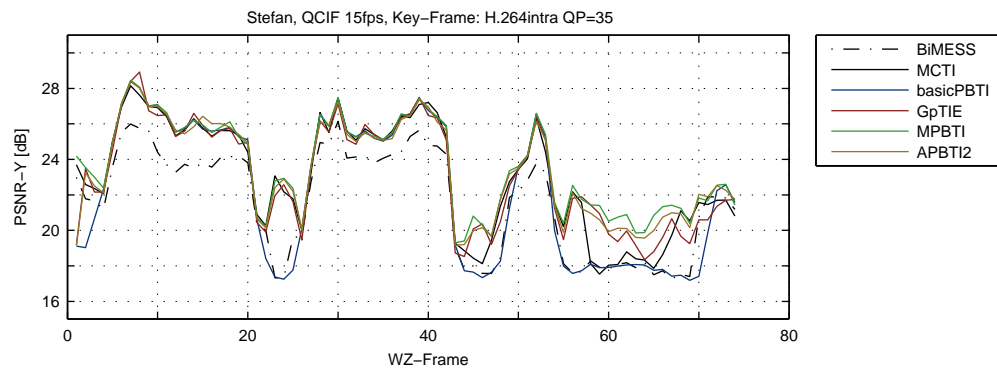


Abbildung C.13.: Qualität der Seiteninformation per WZ-Frame, **Stefan**, QCIF, 15 fps, Key-Frames: H.264intra, $QP = 35$

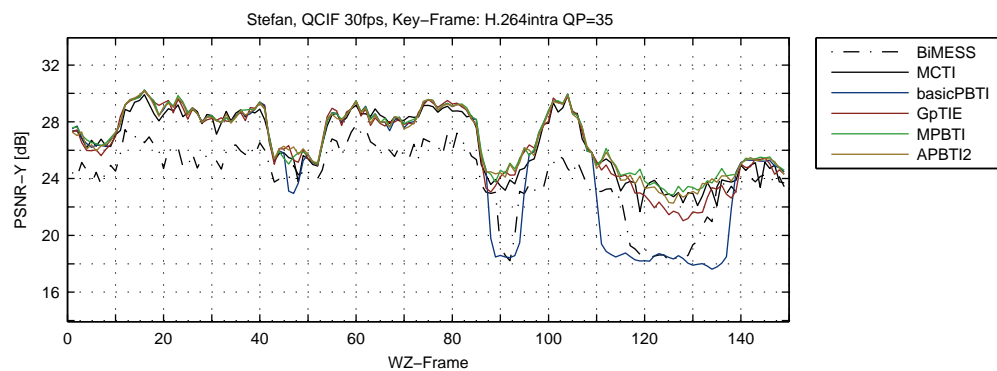


Abbildung C.14.: Qualität der Seiteninformation per WZ-Frame, **Stefan**, QCIF, 30 fps, Key-Frames: H.264intra, $QP = 35$

Bus – CIF

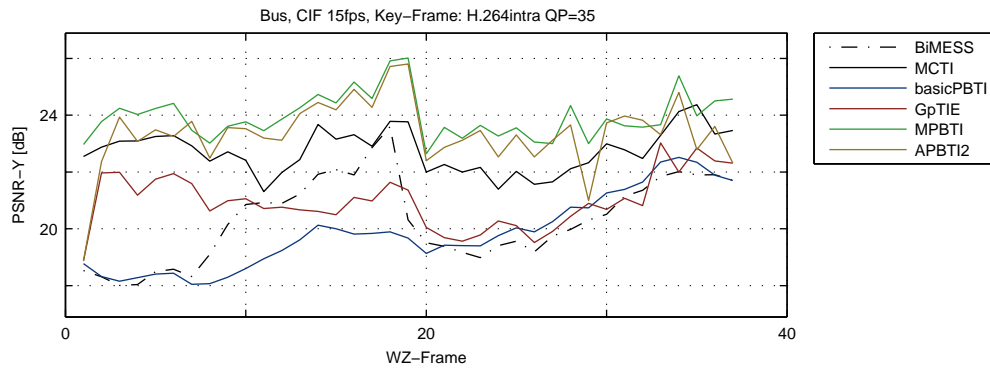


Abbildung C.15.: Qualität der Seiteninformation per WZ-Frame, Bus, CIF, 15 fps, Key-Frames: H.264intra, $QP = 35$

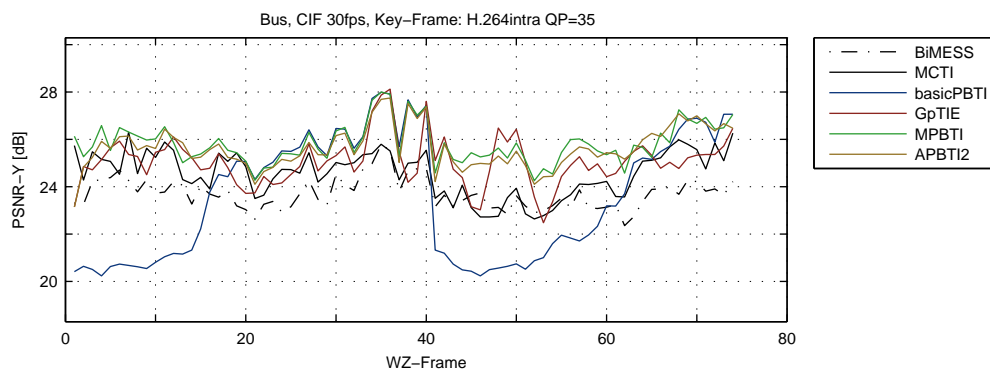


Abbildung C.16.: Qualität der Seiteninformation per WZ-Frame, Bus, CIF, 30 fps, Key-Frames: H.264intra, $QP = 35$

SFlowG – SIF

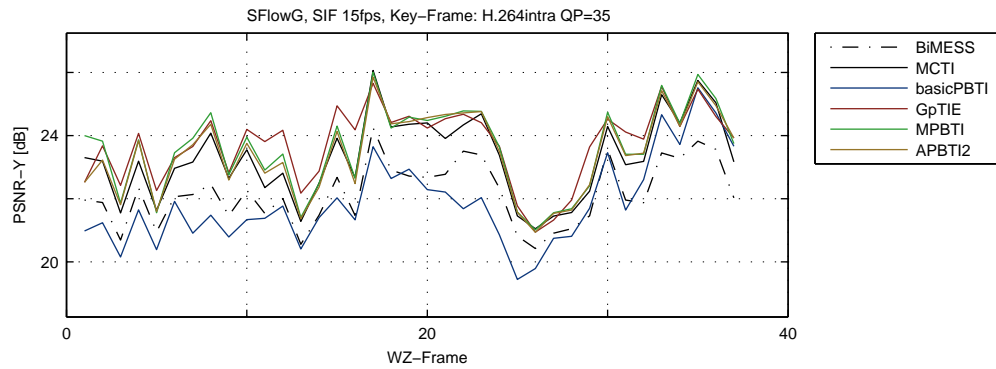


Abbildung C.17.: Qualität der Seiteninformation per WZ-Frame, SFlowG, SIF, 15 fps, Key-Frames: H.264intra, $QP = 35$

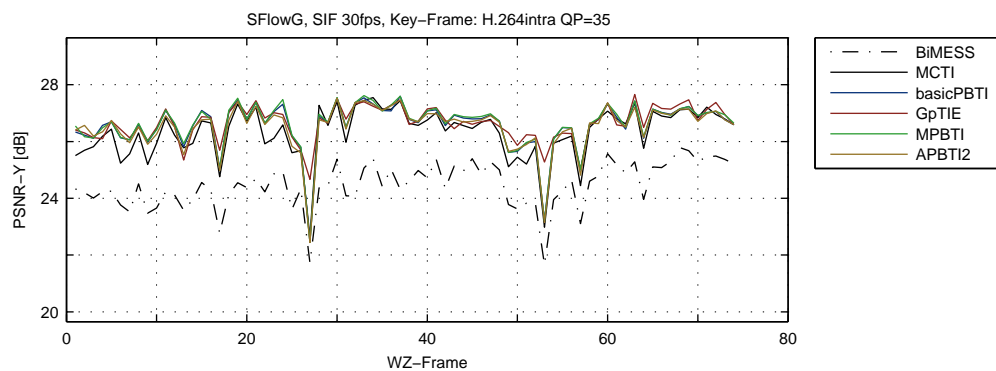


Abbildung C.18.: Qualität der Seiteninformation per WZ-Frame, SFlowG, SIF, 30 fps, Key-Frames: H.264intra, $QP = 35$

C.3. Differenz PSNR-Y Key-Frames und Seiteninformation

Die Differenz zwischen dem PSNR der Seiteninformation $PSNR - Y_{SI}$ und dem der H.264intra codierten Key-Frames $PSNR - Y_{Key}$ in Abhängigkeit von der Qualität der Key-Frames ist im Folgenden dargestellt. Der Quantisierungsparameter wird im Bereich $QP = 20, 25, \dots, 50$ variiert.

Akiyo – QCIF

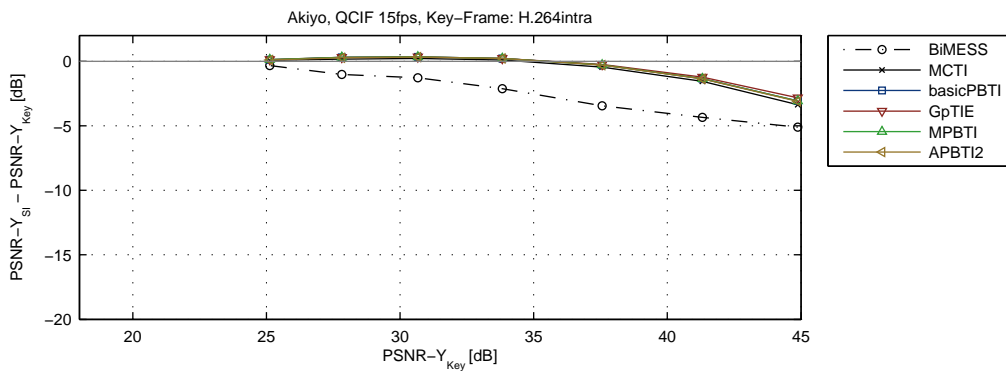


Abbildung C.19.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, Akiyo, QCIF, 15 fps, Key-Frames: H.264intra

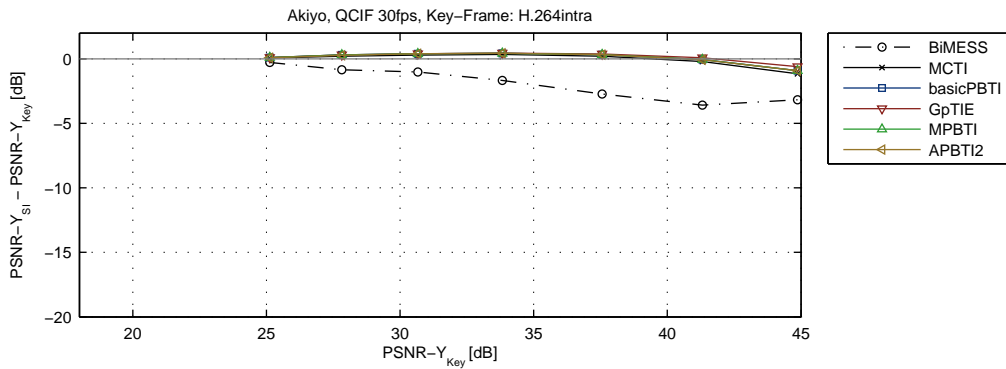


Abbildung C.20.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, Akiyo, QCIF, 30 fps, Key-Frames: H.264intra

Coastguard – QCIF

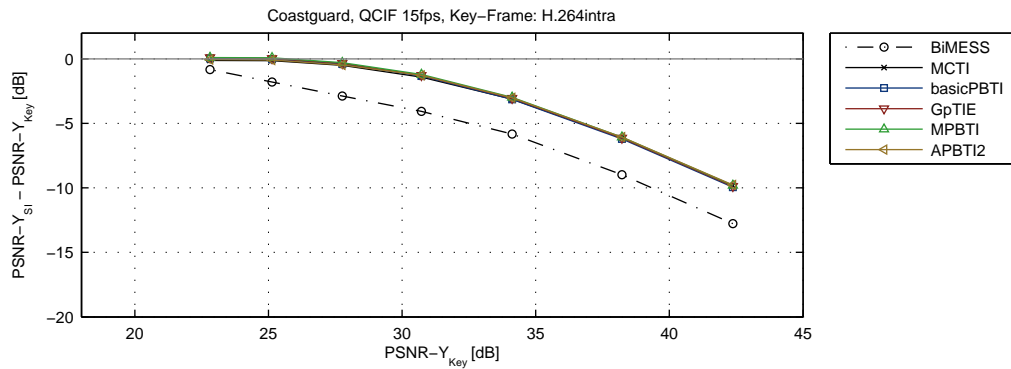


Abbildung C.21.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, Coastguard, QCIF, 15 fps, Key-Frames: H.264intra

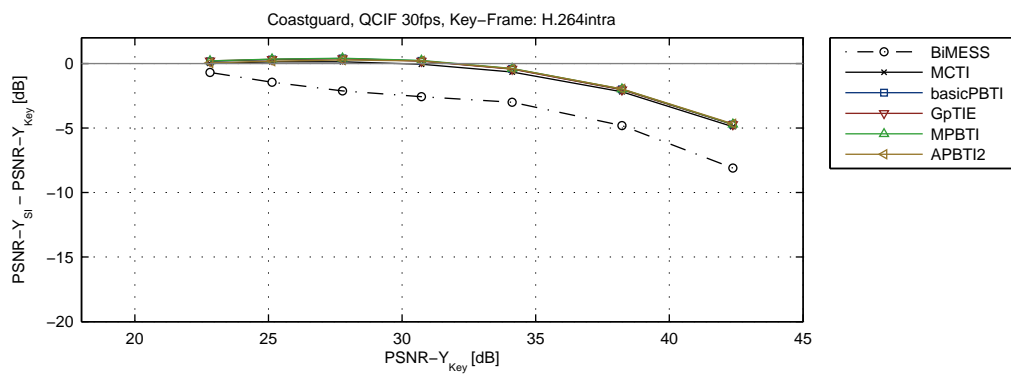


Abbildung C.22.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, Coastguard, QCIF, 30 fps, Key-Frames: H.264intra

Foreman – QCIF

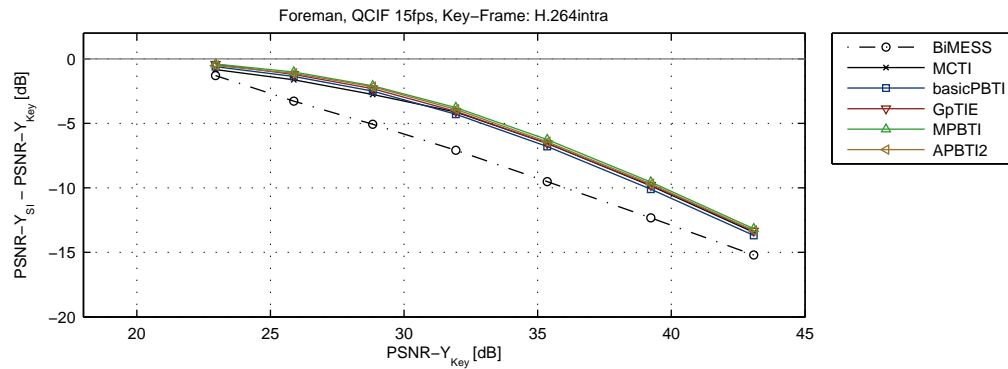


Abbildung C.23.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, Foreman, QCIF, 15 fps, Key-Frames: H.264intra

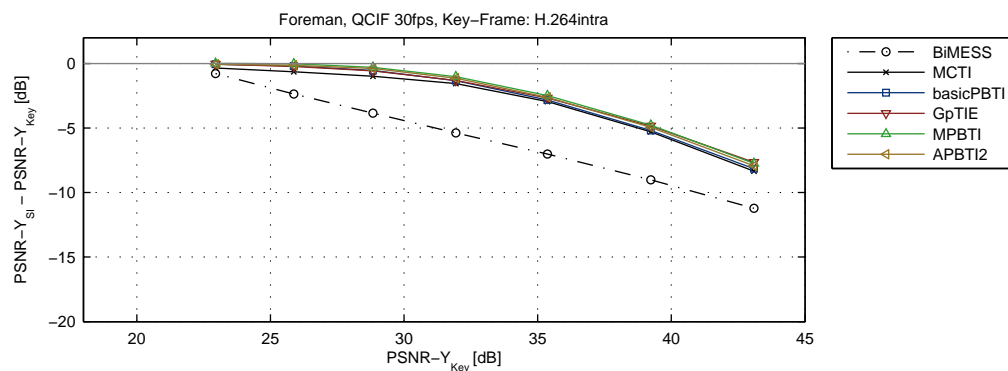


Abbildung C.24.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, Foreman, QCIF, 30 fps, Key-Frames: H.264intra

Hall – QCIF

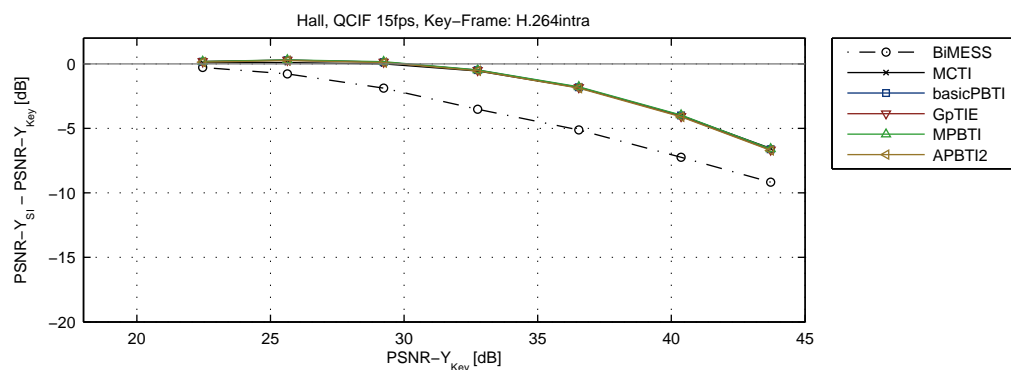


Abbildung C.25.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, Hall, QCIF, 15 fps, Key-Frames: H.264intra

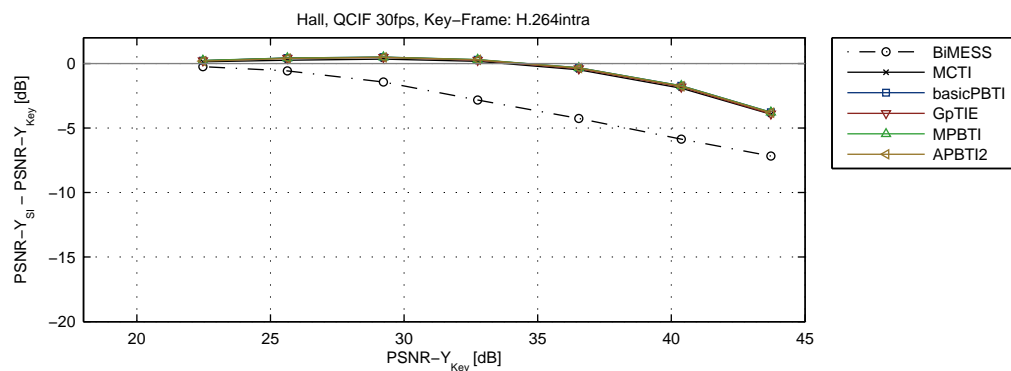


Abbildung C.26.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, Hall, QCIF, 30 fps, Key-Frames: H.264intra

Mother and Daughter – QCIF

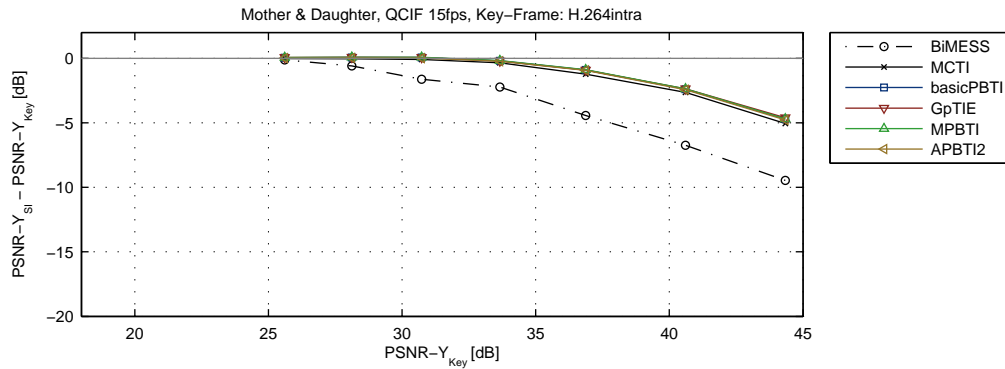


Abbildung C.27.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, Mother and Daughter, QCIF, 15 fps, Key-Frames: H.264intra

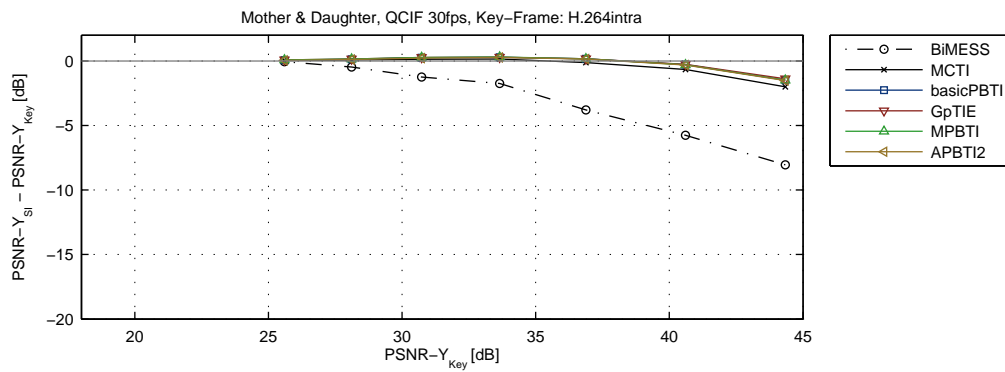


Abbildung C.28.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, Mother and Daughter, QCIF, 30 fps, Key-Frames: H.264intra

Soccer – QCIF

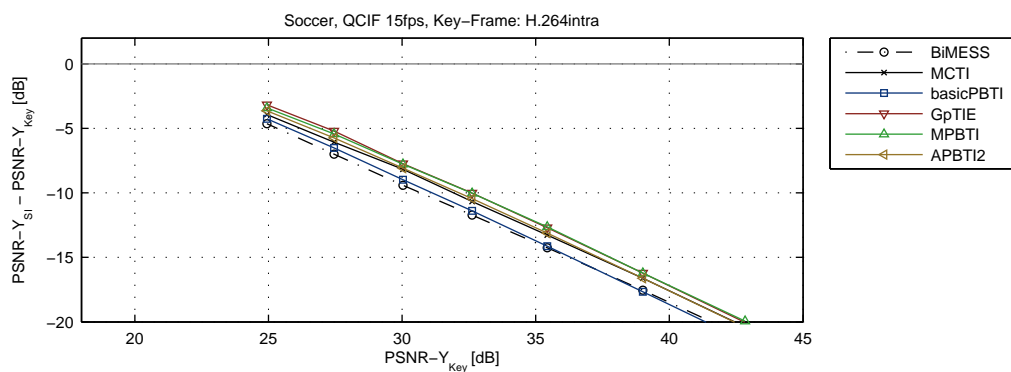


Abbildung C.29.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, **Soccer**, QCIF, 15 fps, Key-Frames: H.264intra

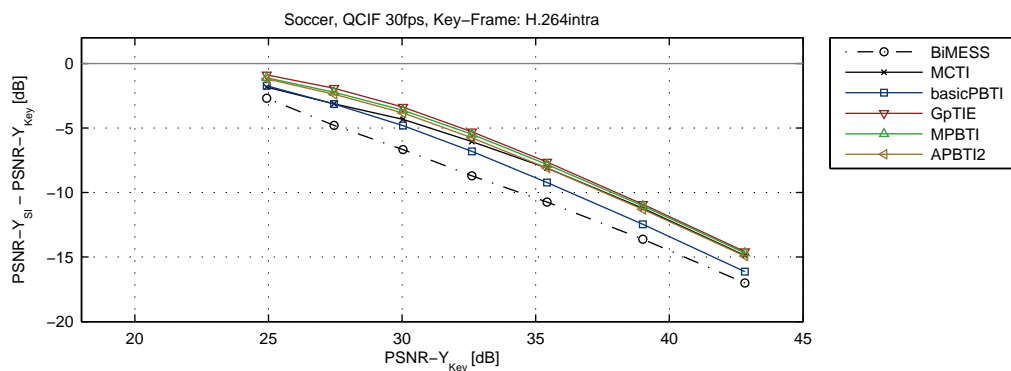


Abbildung C.30.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, **Soccer**, QCIF, 30 fps, Key-Frames: H.264intra

Stefan – QCIF

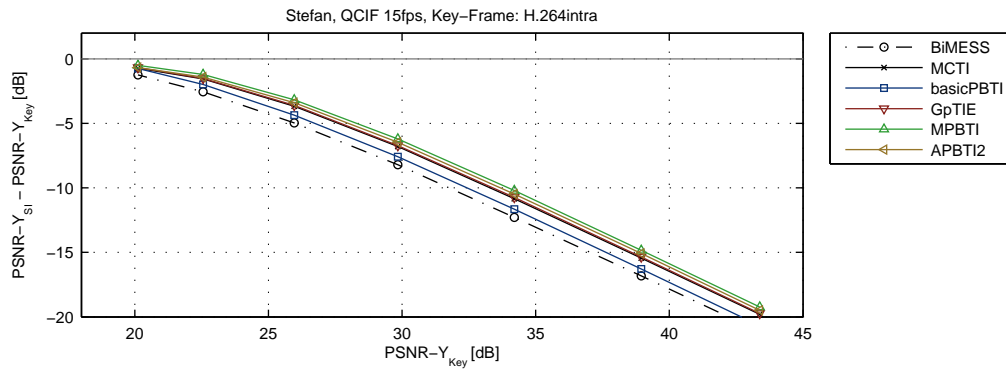


Abbildung C.31.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, **Stefan**, QCIF, 15 fps, Key-Frames: H.264intra

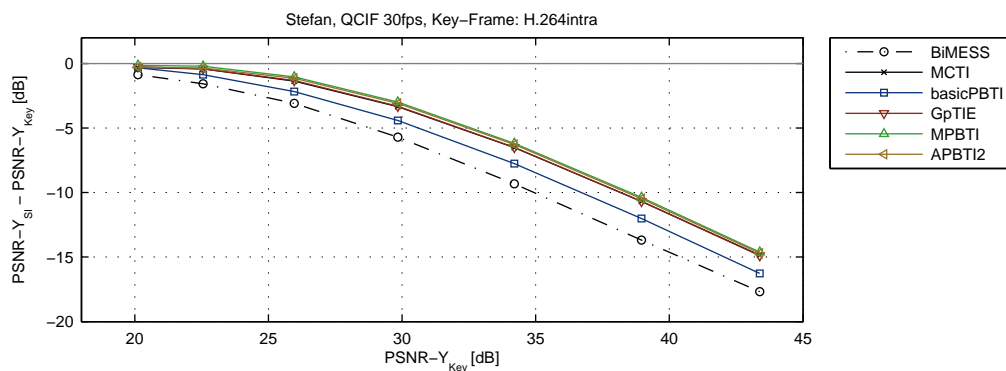


Abbildung C.32.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, **Stefan**, QCIF, 30 fps, Key-Frames: H.264intra

Bus – CIF

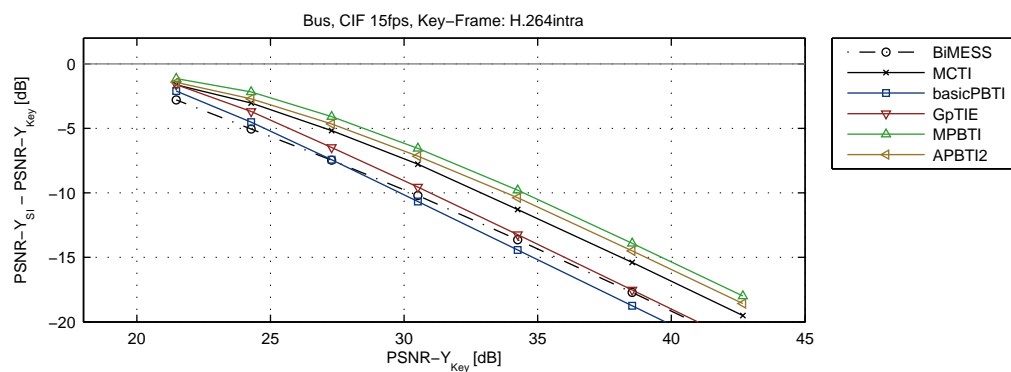


Abbildung C.33.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, Bus, CIF, 15 fps, Key-Frames: H.264intra

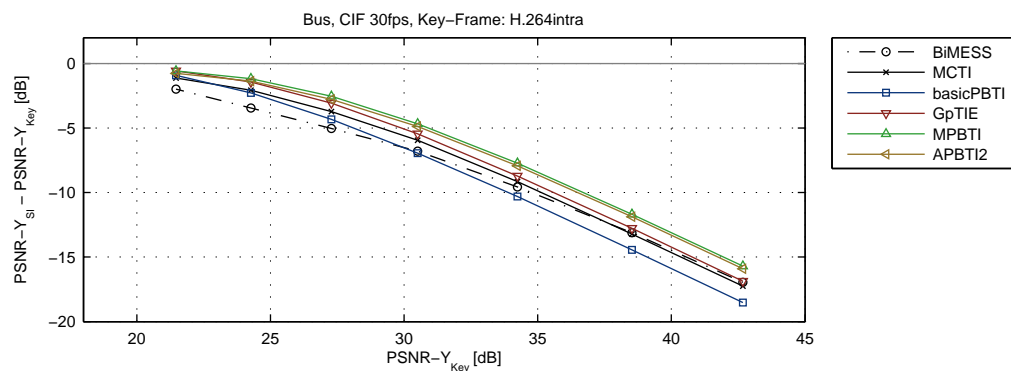


Abbildung C.34.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, Bus, CIF, 30 fps, Key-Frames: H.264intra

SFlowG – SIF

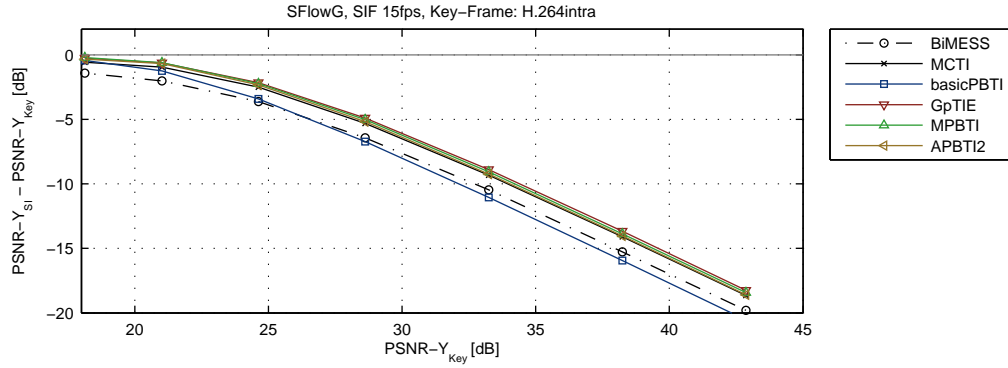


Abbildung C.35.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, SFlowG, SIF, 15 fps, Key-Frames: H.264intra

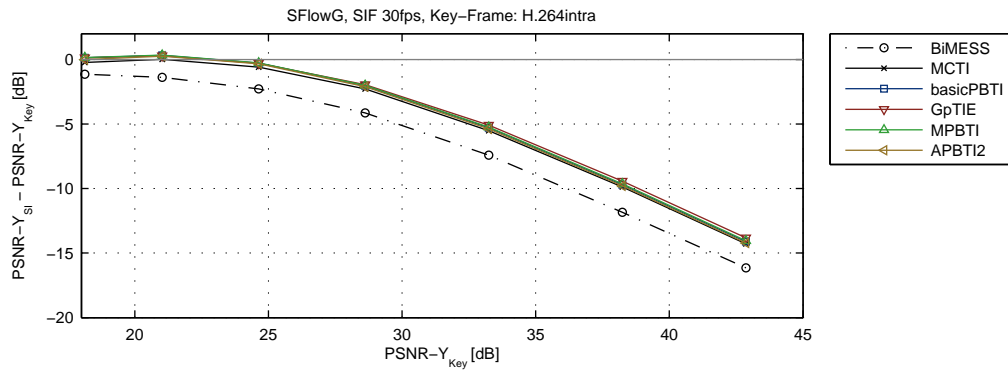


Abbildung C.36.: Differenz zwischen Qualität der Seiteninformation und Key-Frames, SFlowG, SIF, 30 fps, Key-Frames: H.264intra



RD-Performanz des Gesamtsystems

Dieser Anhang zeigt die Rate-Distortion Performanz des entwickelten Codierungssystems. Es umfasst die Auswahl der Codierungsparameter und Referenzen sowie die Darstellung der RD-Performanz in Abhängigkeit vom verwendeten Verfahren zur Generierung der Seiteninformation als auch in Abhängigkeit vom verwendeten Codierungsverfahren für die Key-Frames.

D.1. Parameter

Die folgenden Parameter zur Codierung der Testsequenzen mit dem entwickelten DVC Codec *oDeViCe* wurden verwendet:

- GOP-Länge von 2 Frames (K-W-K-W-K-W...)
- Key-Frames codiert mit H.264intra (JM-18.0, [Wiegand u. a. 2003][H.264 2012]) bzw. HEVCintra (HM-7.2, [Bross u. a. 2012][HEVC 2012][Meur 2011])
- zufällige Auswahlreihenfolge der Pixel
- offline Schätzung des Modellparameters λ auf Frame-Ebene (ein Parameter für den gesamten Frame)
- MMSE Rekonstruktion
- Turbocode basierte Slepian-Wolf Decodierung
- Prüfung auf erfolgreiche Decodierung mittels CRC

- Bitebenen-weise Decodierung beginnend vom MSB zum LSB
- Schätzung der a-priori Information am Decoder (AIE)
- volle Nutzung des Rückkanals zur Decoder-seitigen Ratenkontrolle

D.2. Rate-Distortion Punkte

Die Rate-Distortion Punkte für die Codierung der Testsequenzen wurden anhand der Seiteninformationsqualität, in Abhängigkeit von der Rekonstruktionsqualität der Key-Frames, bestimmt. Je geringer die Qualität der Seiteninformation $PSNR - Y_{SI}$ im Vergleich zu der Qualität der Key-Frames $PSNR - Y_{Key}$ ist, desto größer wurde der Quantisierungsparameter M für die WZ-Frames gewählt (Abb. D.1).

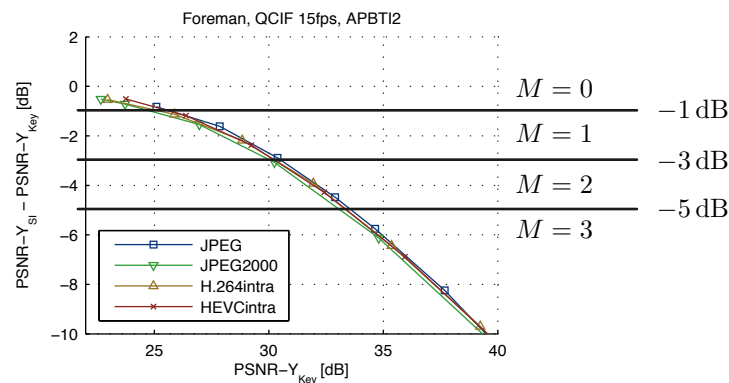


Abbildung D.1.: Auswahl der RD-Punkte (QP, M) anhand der Differenz zwischen Key-Frame PSNR und PSNR der Seiteninformation

Tabelle D.1 (S. 266) zeigt die so ausgewählten RD-Punkte als Tupel (QP, M), aus Quantisierungsparameter QP der Key-Frames und Quantisierungsparameter M der WZ-Frames.

D.3. Referenzen

Zur Einordnung der Leistungsfähigkeit des Videocodecs *oDeViCe* wurden folgende Referenzverfahren aus dem Bereich der konventionellen Videocodierung gewählt. Diese Referenzverfahren verwenden keine Bewegungsschätzung, wodurch die Komplexität des Encoders wesentlich reduziert wird.

H.264intra — Alle Frames der Videosequenz werden als I-Bilder mit der Referenzsoftware JM-18.0 [Wiegand u. a. 2003][H.264 2012] codiert. Dabei wird das Baseline Profile angewendet.

H.264IBIB — Unter Verwendung der Referenzsoftware JM-18.0 [Wiegand u. a. 2003][H.264 2012] werden die Videosequenzen in der GOP-Struktur I-B-I-B-... codiert. Dabei wird am Encoder keine Bewegungsschätzung und -kompensation für die B-Bilder durchgeführt, womit die zeitlichen Abhängigkeiten, mit Rücksicht auf die Komplexität, nur begrenzt ausgenutzt werden können.

HEVCintra — Die Testsequenzen werden mit der Referenzsoftware HM-7.2 [Bross u. a. 2012][HEVC 2012], des noch nicht standardisierten Codierungsverfahren HEVC, komprimiert. Es werden ausschließlich I-Bilder verwendet.

D.4. RD-Performanz

Akiyo – QCIF

- Aufgrund geringer Bewegung sind nur geringe Verbesserung der RD-Performanz mit GpTIE erreicht
- RD-Performanz zeigt wesentlichen Gewinn gegenüber H.264intra
- Schwankungen zwischen aufeinanderfolgenden Frames geringer als 1 dB

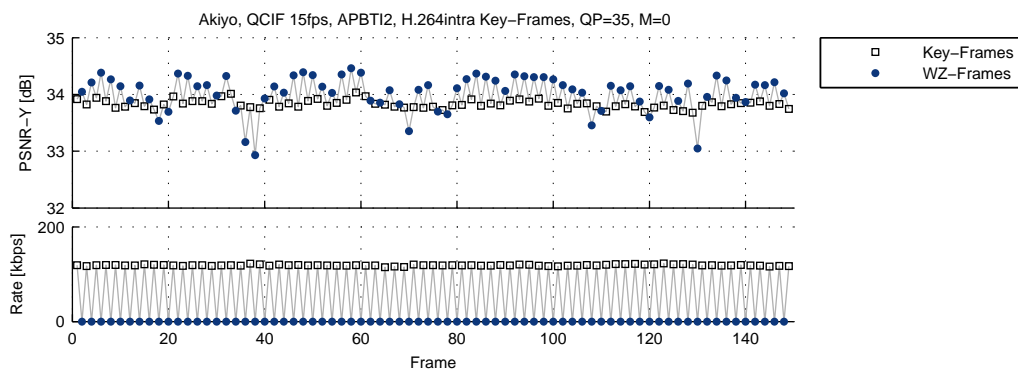


Abbildung D.2.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Akiyo, QCIF, 15 fps

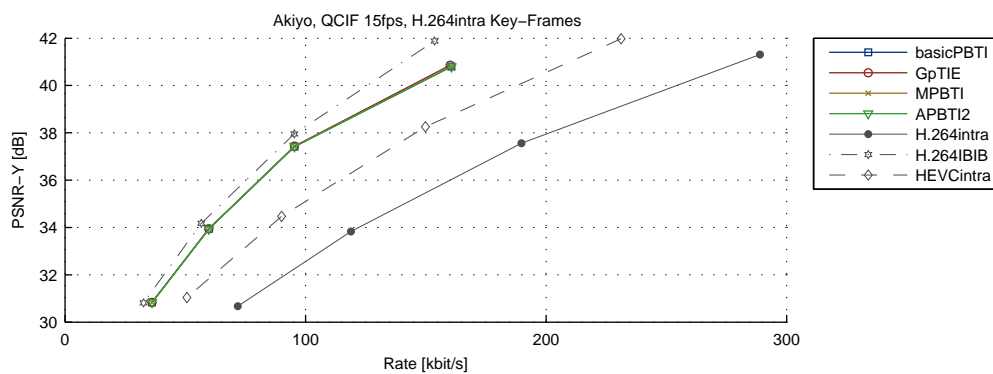


Abbildung D.3.: RD-Performanz des Gesamtcodecs, Akiyo, QCIF, 15 fps

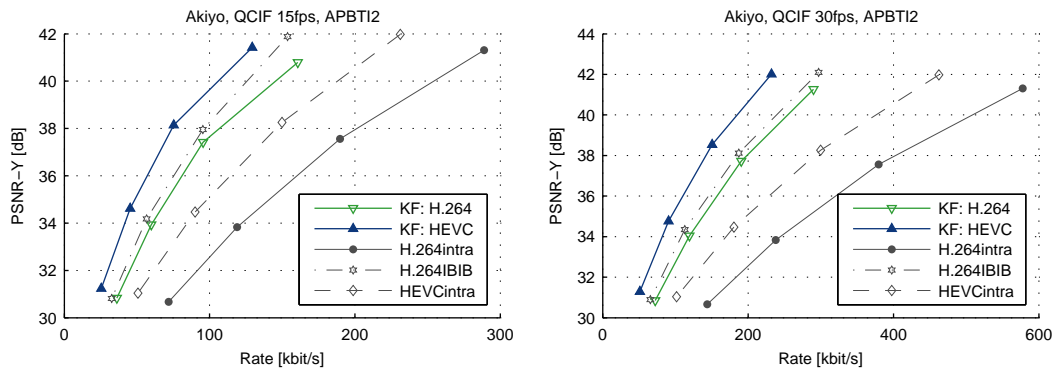


Abbildung D.4.: RD-Performanz des Gesamtcodecs, Akiyo, QCIF

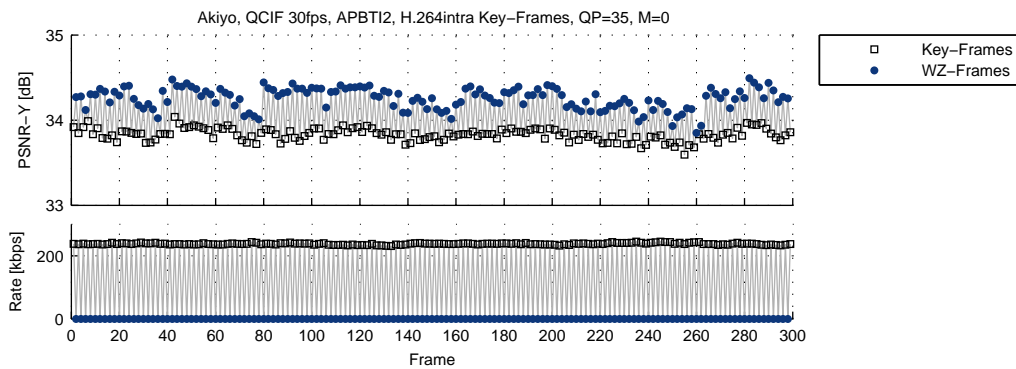


Abbildung D.5.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Akiyo, QCIF, 30 fps

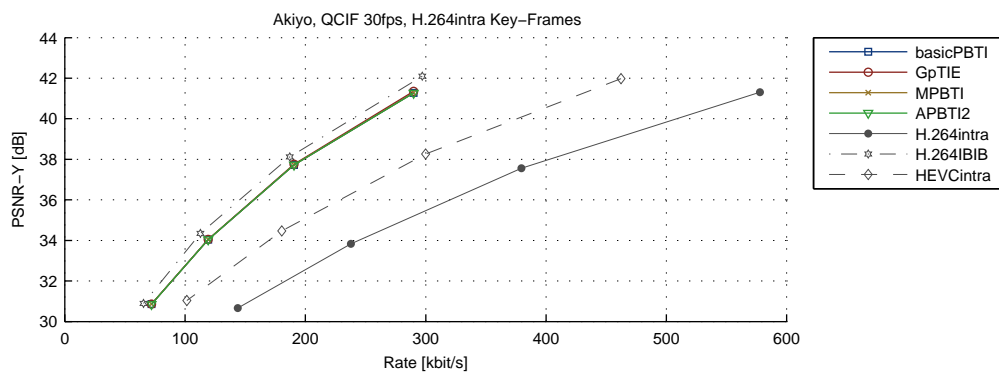


Abbildung D.6.: RD-Performanz des Gesamtcodecs, Akiyo, QCIF, 30 fps

Coastguard – QCIF

- Wahl der Seiteninformationsgenerierung hat aufgrund der einfachen Bewegungsstruktur wenig Einfluss auf die RD-Performanz
- RD-Performanz ähnlich hoch wie bei H.264IBIB
- teilweise starke Qualitätsänderungen aufeinanderfolgender Frames (bis 8 dB)
- überwiegend geringe Schwankungen des PSNR zwischen den Frames

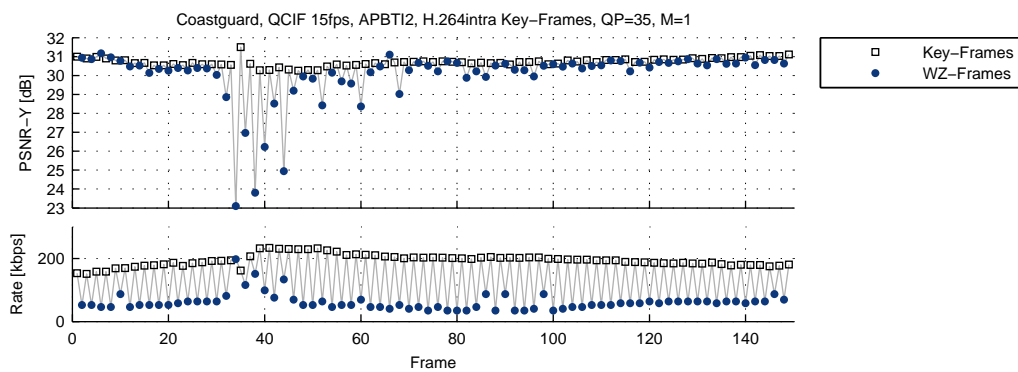


Abbildung D.7.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Coastguard, QCIF, 15 fps

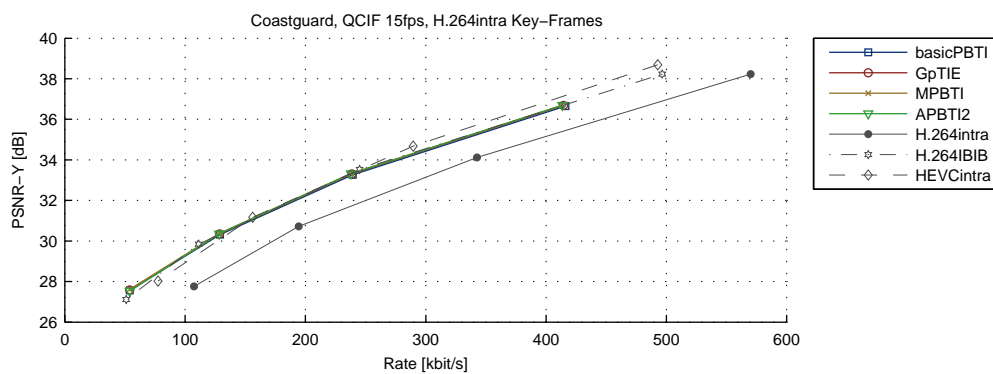


Abbildung D.8.: RD-Performanz des Gesamtcodecs, Coastguard, QCIF, 15 fps

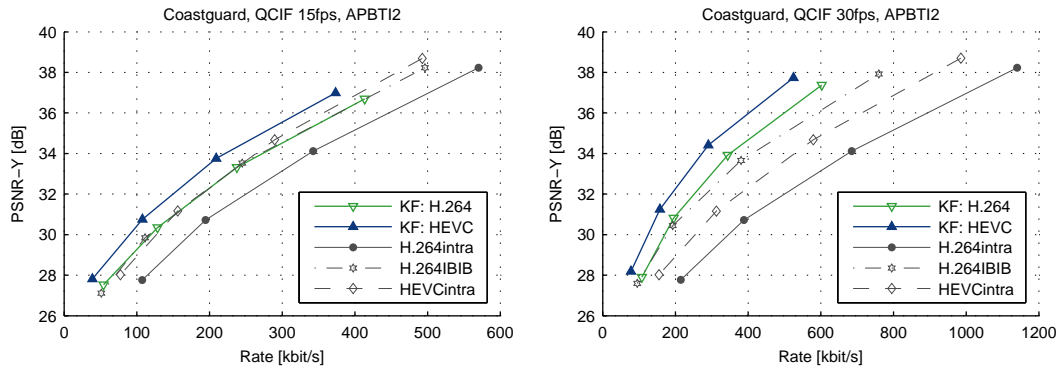


Abbildung D.9.: RD-Performanz des Gesamtcodecs, Coastguard, QCIF

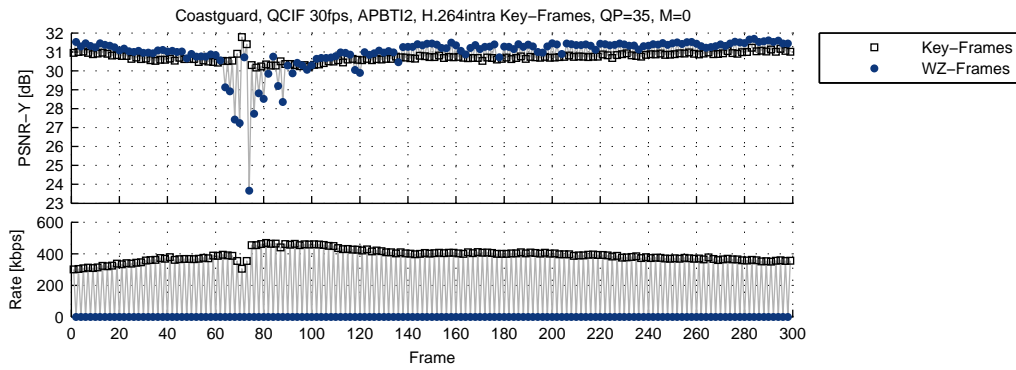


Abbildung D.10.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Coastguard, QCIF, 30 fps

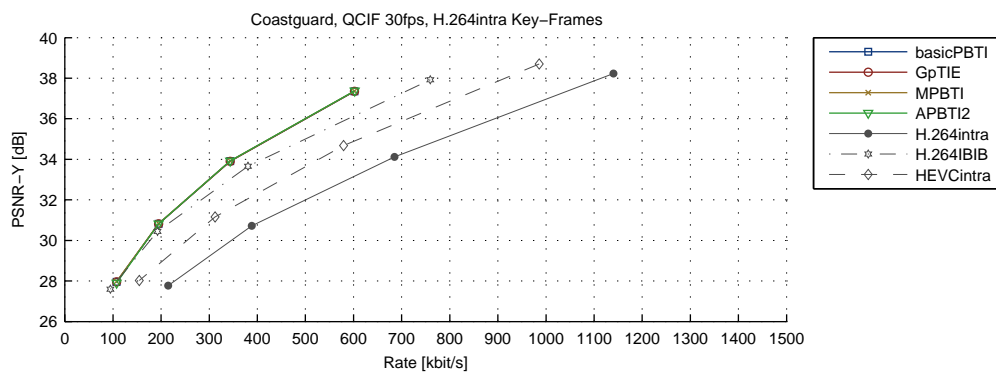


Abbildung D.11.: RD-Performanz des Gesamtcodecs, Coastguard, QCIF, 30 fps

Foreman – QCIF

- Verfahren, wie GpTIE, MPBTI oder APBTI2, welche die globale Bewegung zur Generierung der Seiteninformation berücksichtigen, zeigen eine Steigerung der RD-Performanz
- bei geringer Framerate 15 fps zeigen die WZ-Frames eine geringere Rekonstruktionsqualität und höhere Datenrate als die Key-Frames, was zu einer schlechten RD-Performanz führt
- bei hoher Framerate 30 fps wird eine RD-Performanz ähnlich HEVCintra erreicht
- stärkere Einbrüche der Qualität zwischen den Key- und WZ-Frames bei der 15 fps Sequenz

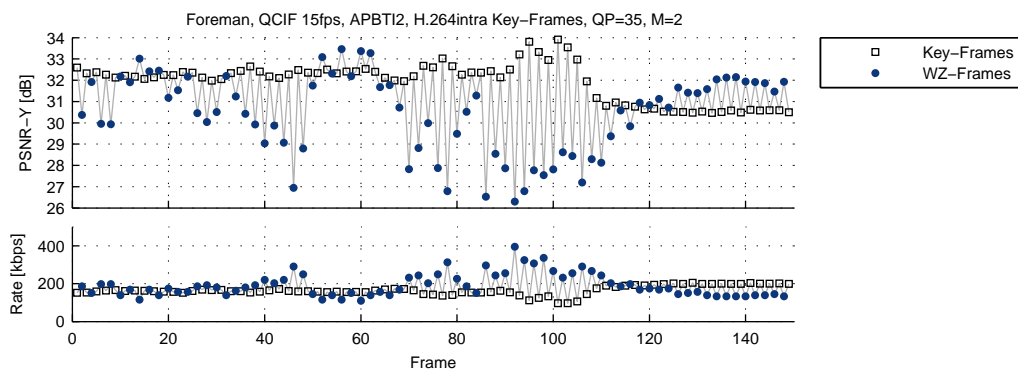


Abbildung D.12.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Foreman, QCIF, 15 fps

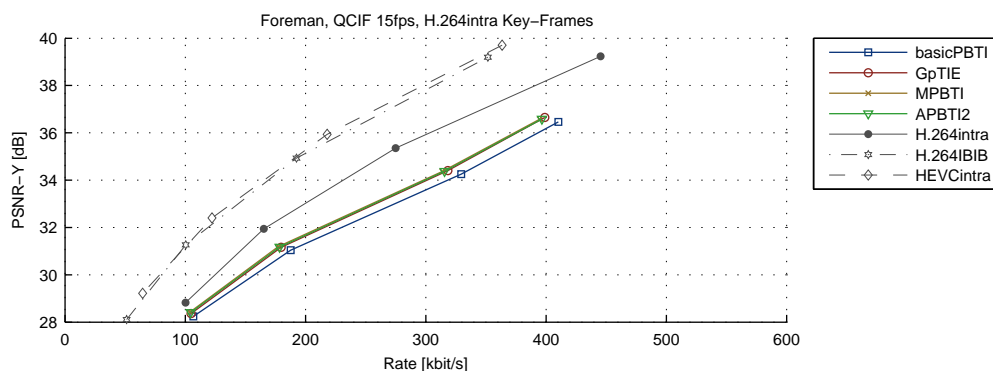


Abbildung D.13.: RD-Performanz des Gesamtcodecs, Foreman, QCIF, 15 fps

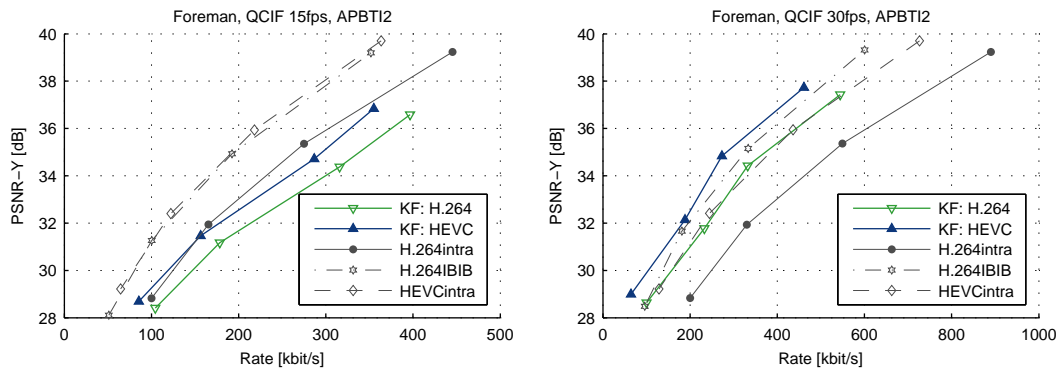


Abbildung D.14.: RD-Performanz des Gesamtcodecs, Foreman, QCIF

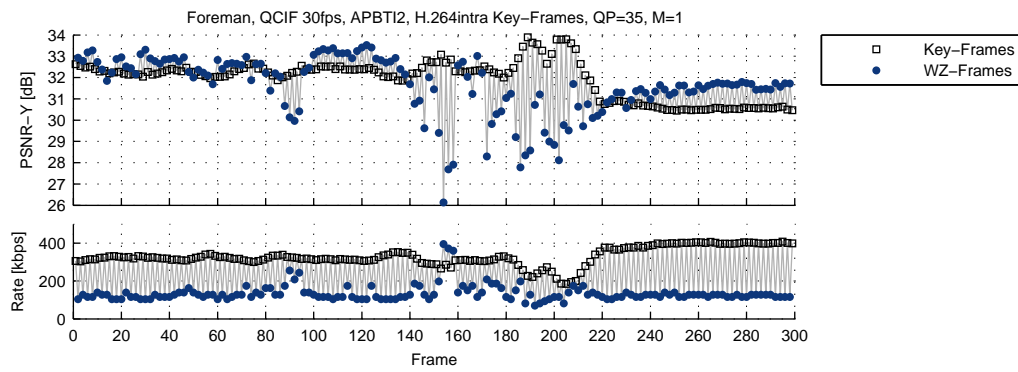


Abbildung D.15.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Foreman, QCIF, 30 fps

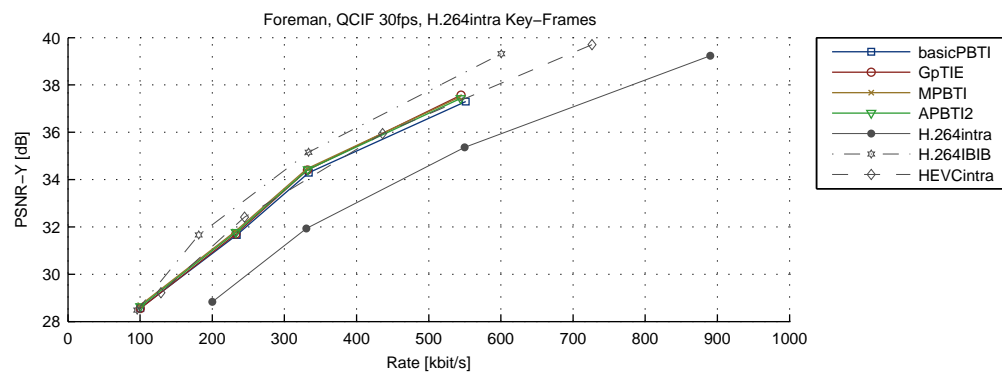


Abbildung D.16.: RD-Performanz des Gesamtcodecs, Foreman, QCIF, 30 fps

Ha11 – QCIF

- Aufgrund der sehr einfachen Bewegungsstruktur fällt die RD-Performanz für alle betrachteten Seiteninformationsgenerierungsverfahren hoch aus
- die Qualitätsschwankungen zwischen den Frames sind mit 2 – 3 dB moderat

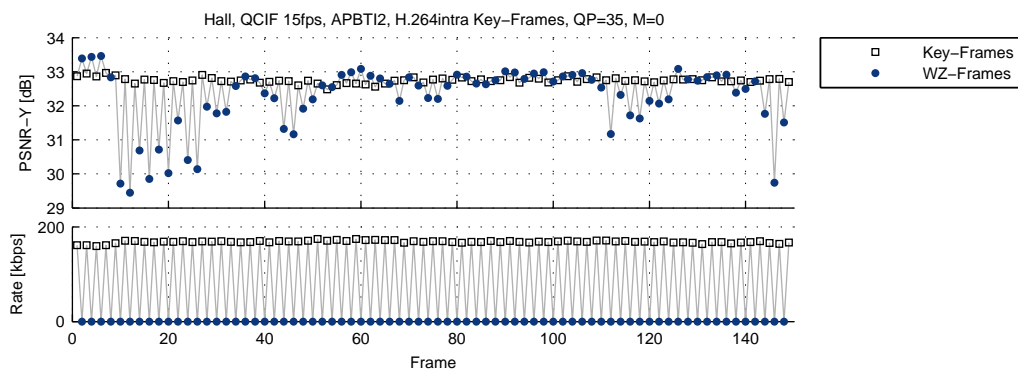


Abbildung D.17.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Ha11, QCIF, 15 fps

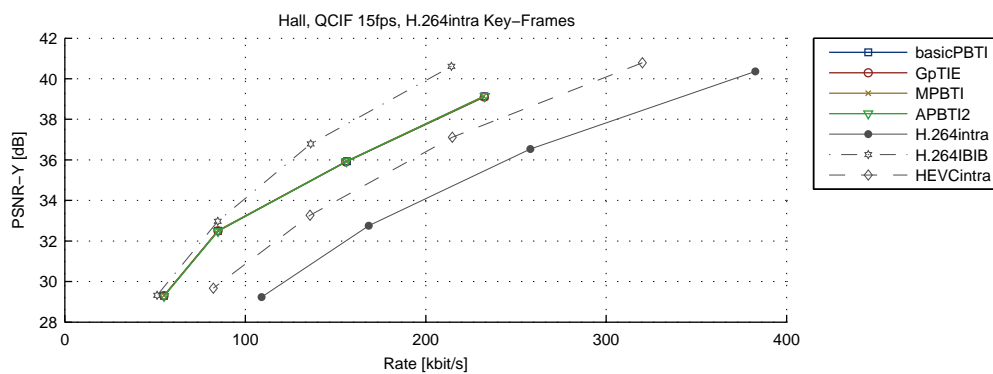


Abbildung D.18.: RD-Performanz des Gesamtcodecs, Ha11, QCIF, 15 fps

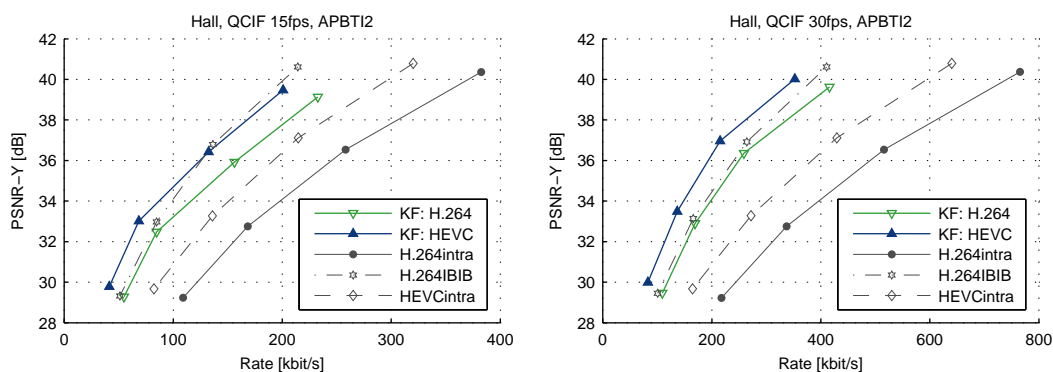


Abbildung D.19.: RD-Performanz des Gesamtcodecs, Ha11, QCIF

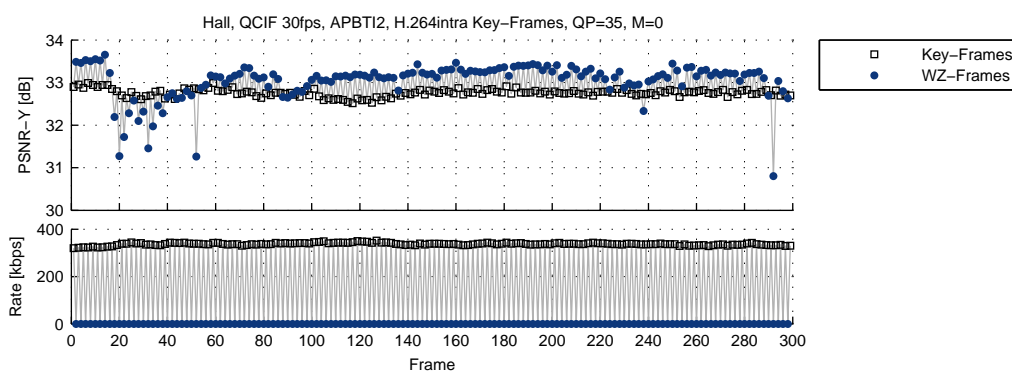


Abbildung D.20.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Ha11, QCIF, 30 fps

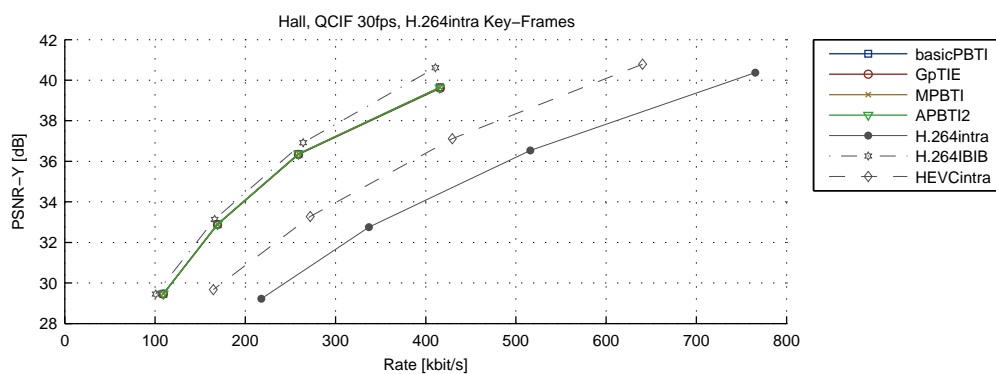


Abbildung D.21.: RD-Performanz des Gesamtcodecs, Ha11, QCIF, 30 fps

Mother and Daughter – QCIF

- Aufgrund der geringen Bewegung ist die RD-Performanz nicht vom verwendeten Verfahren zur Generierung der Seiteninformation abhängig
- im Allgemeinen wird eine hohe RD-Performanz erreicht
- teilweise Einbrüche der Qualität der WZ-Frames von bis zu 7 dB, in weiten Bereichen jedoch Änderungen von nur ± 0.5 dB

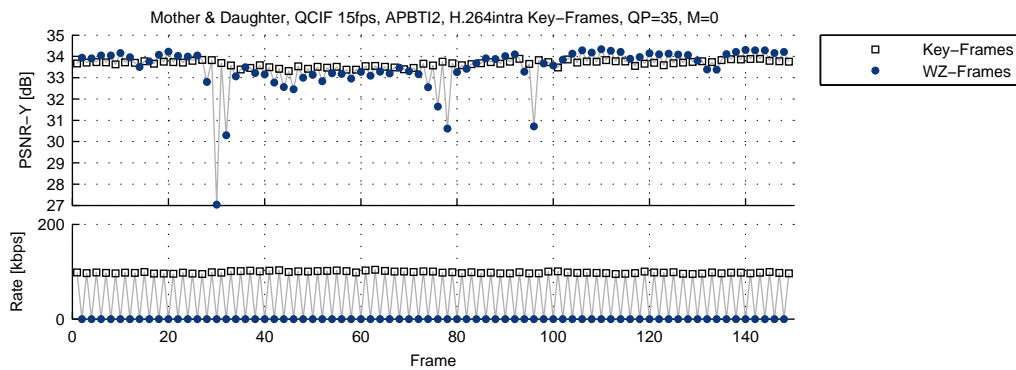


Abbildung D.22.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Mother and Daughter, QCIF, 15 fps

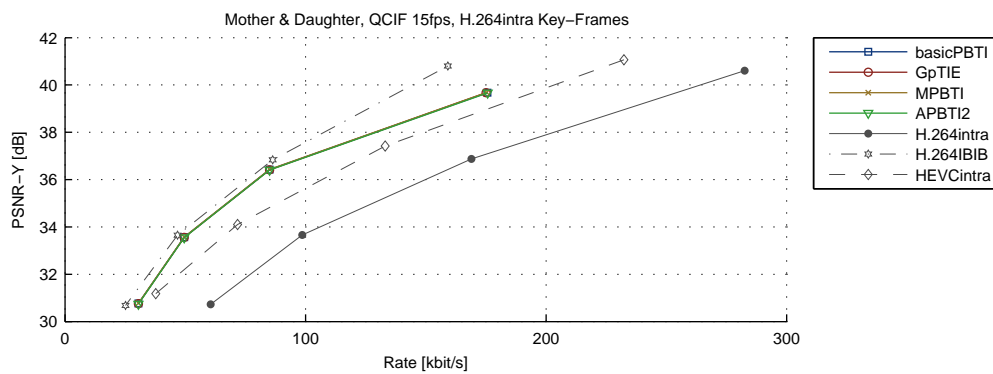


Abbildung D.23.: RD-Performanz des Gesamtcodecs, Mother and Daughter, QCIF, 15 fps

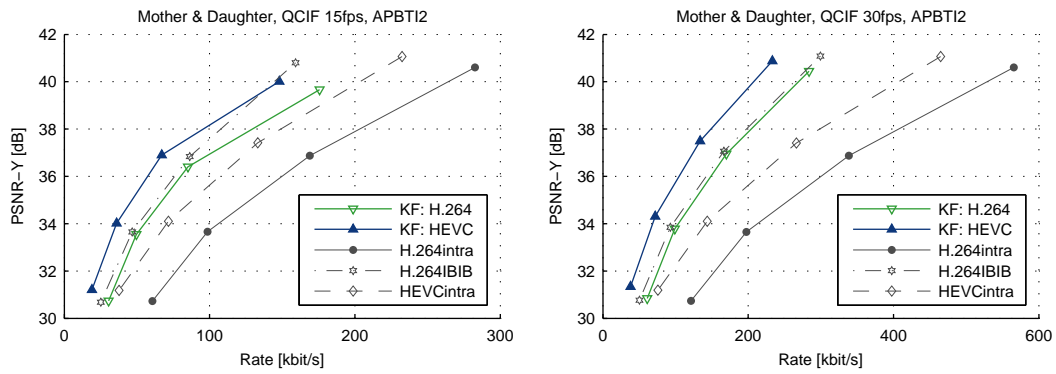


Abbildung D.24.: RD-Performanz des Gesamtcodecs, Mother and Daughter, QCIF

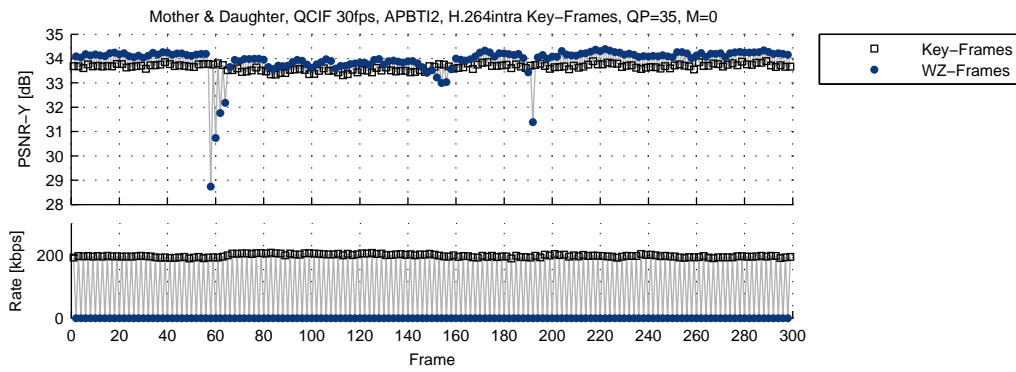


Abbildung D.25.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Mother and Daughter, QCIF, 30 fps

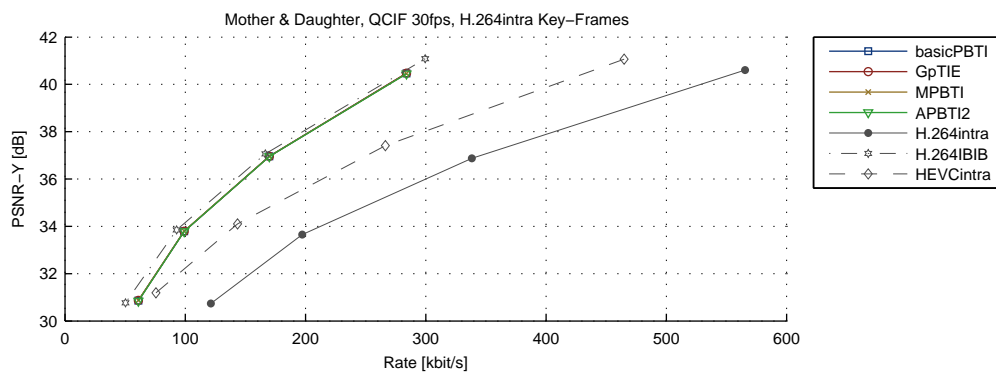


Abbildung D.26.: RD-P. des Gesamtcodecs, Mother and Daughter, QCIF, 30 fps

Soccer – QCIF

- Sequenzen mit sehr komplexer Bewegungssituation im zweiten Teil, wodurch Verfahren wie GpTIE, MPBTI und APBTI2 eine hohe RD-Performanz hervorrufen
- RD-Performanz wesentlich schlechter als die Referenzverfahren, da die zeitliche Korrelation der Sequenz sehr gering ist
- moderate Schwankung der Qualität zwischen den Key- und WZ-Frames von ± 3 dB

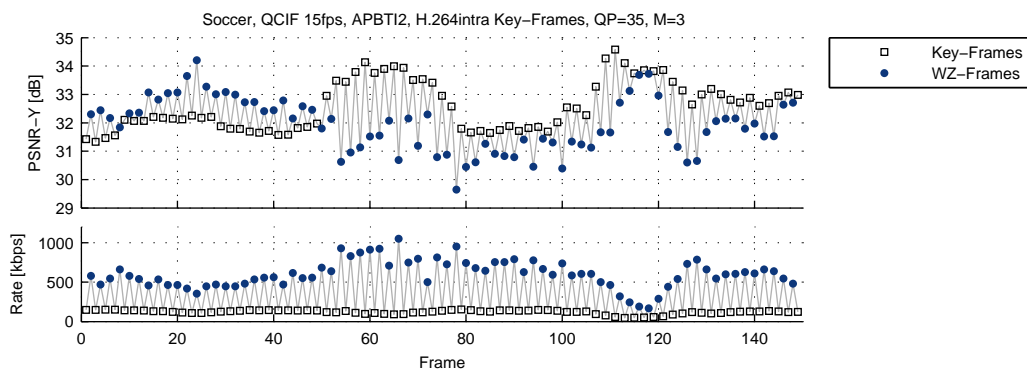


Abbildung D.27.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Soccer, QCIF, 15 fps

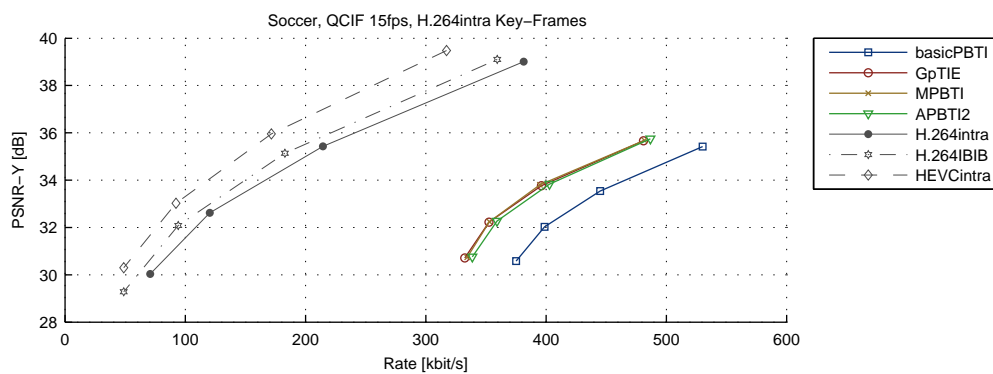


Abbildung D.28.: RD-Performanz des Gesamtcodecs, Soccer, QCIF, 15 fps

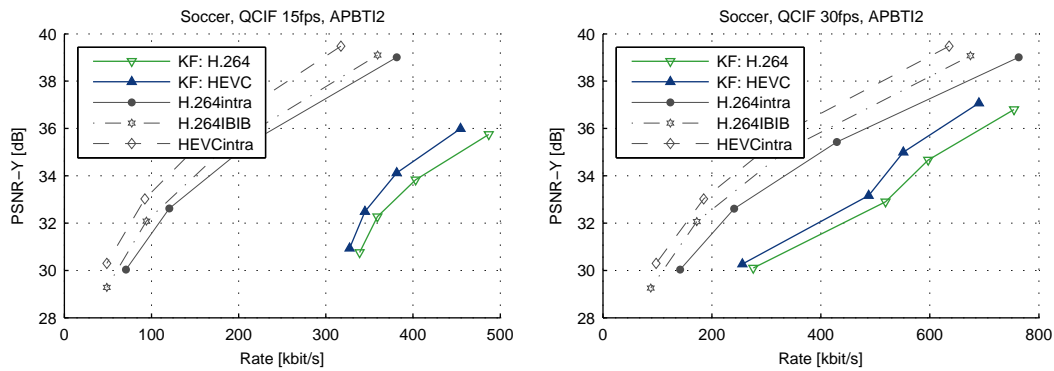


Abbildung D.29.: RD-Performanz des Gesamtcodecs, Soccer, QCIF

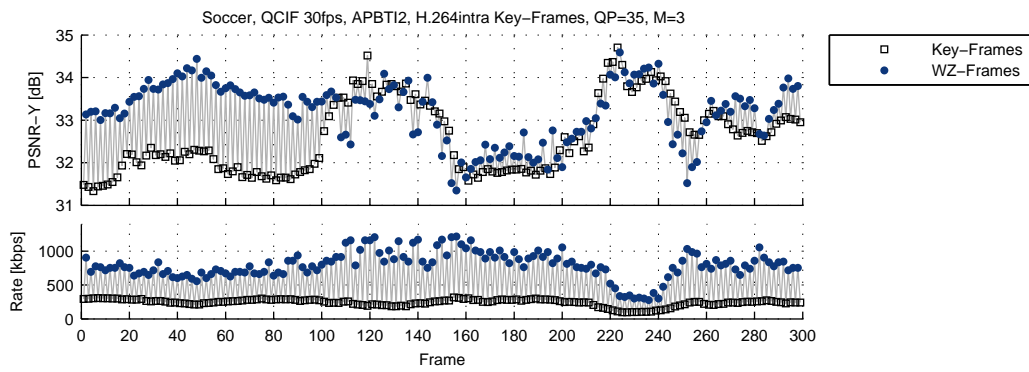


Abbildung D.30.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Soccer, QCIF, 30 fps

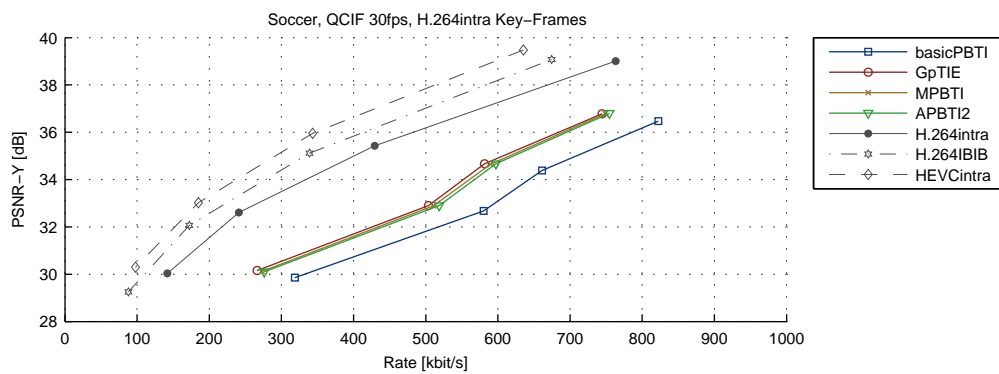


Abbildung D.31.: RD-Performanz des Gesamtcodecs, Soccer, QCIF, 30 fps

Stefan – QCIF

- wesentlich bessere RD-Performanz bei Verfahren, welche die globale Bewegung berücksichtigen (+2 dB)
- die RD-Performanz von H.264intra wird nur für hohe Frameraten von 30 fps erreicht
- moderate Schwankungen der Qualität zwischen den Frames +1.5... – 3 dB

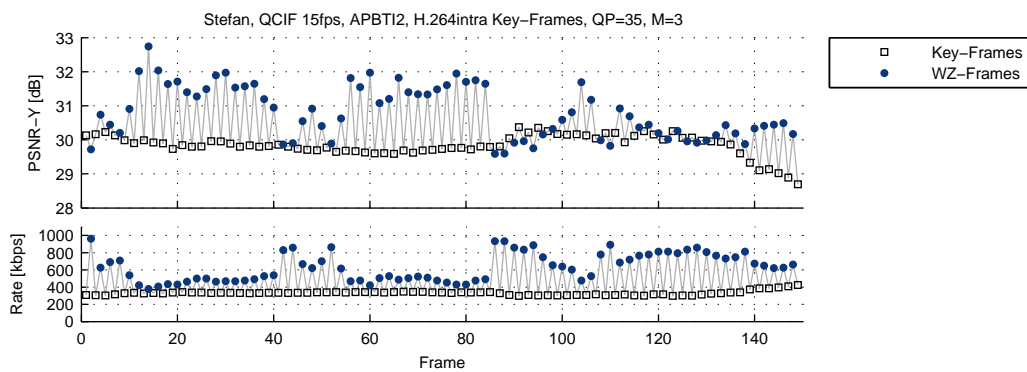


Abbildung D.32.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Stefan, QCIF, 15 fps

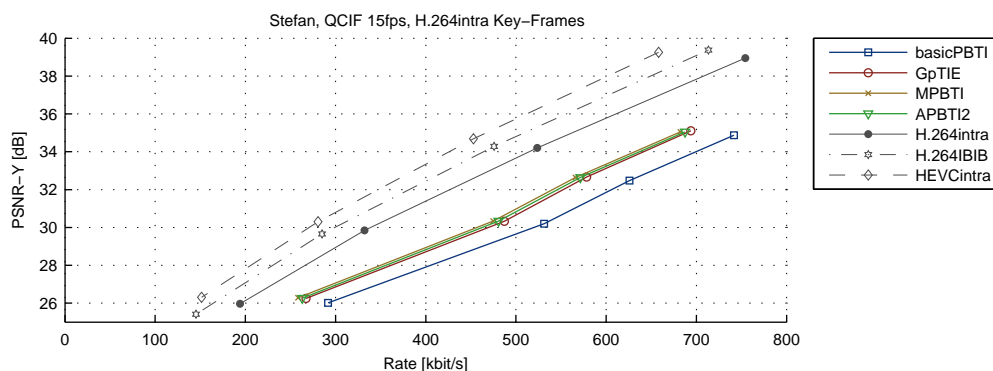
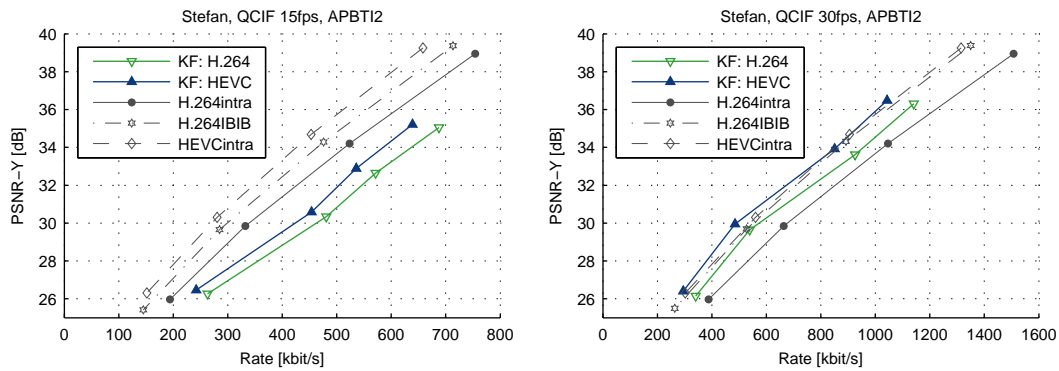
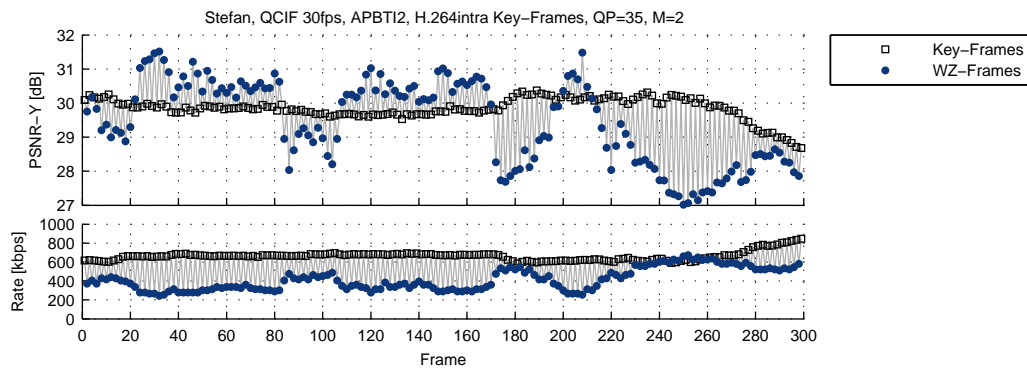
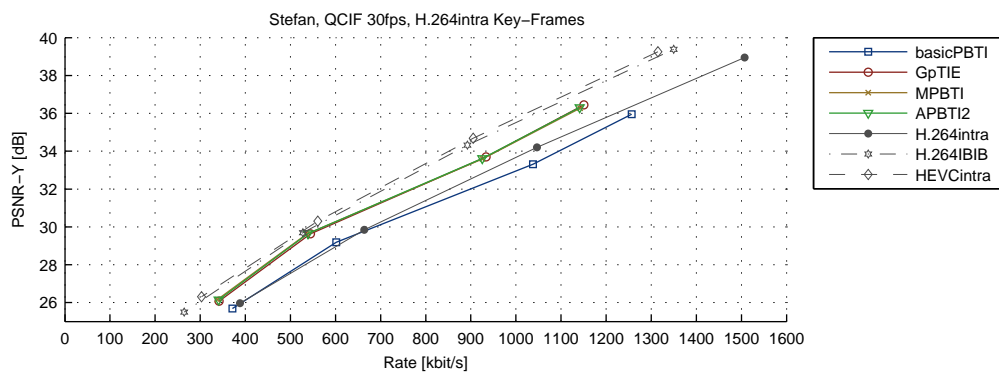


Abbildung D.33.: RD-Performanz des Gesamtcodecs, Stefan, QCIF, 15 fps

Abbildung D.34.: RD-Performanz des Gesamtcodecs, **Stefan**, QCIFAbbildung D.35.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, **Stefan**, QCIF, 30 fpsAbbildung D.36.: RD-Performanz des Gesamtcodecs, **Stefan**, QCIF, 30 fps

Bus – CIF

- wesentliche Verbesserung der RD-Performanz durch die Verwendung von M-PBTI bzw. APBTI2, da diese inhomogene Bewegung berücksichtigen
- RD-Performanz schlechter als H.264intra, bei hoher Framerate 30 fps circa -0.4 dB
- moderate Schwankungen zwischen den Frames ($+1 \dots -1.5$ dB)

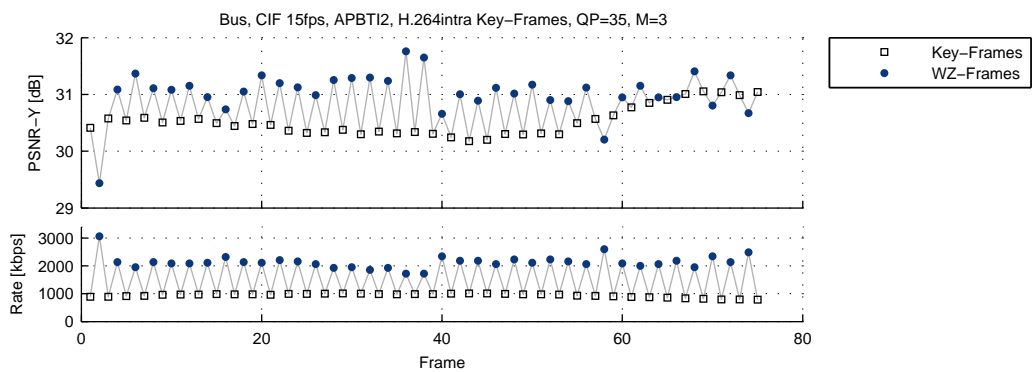


Abbildung D.37.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Bus, CIF, 15 fps

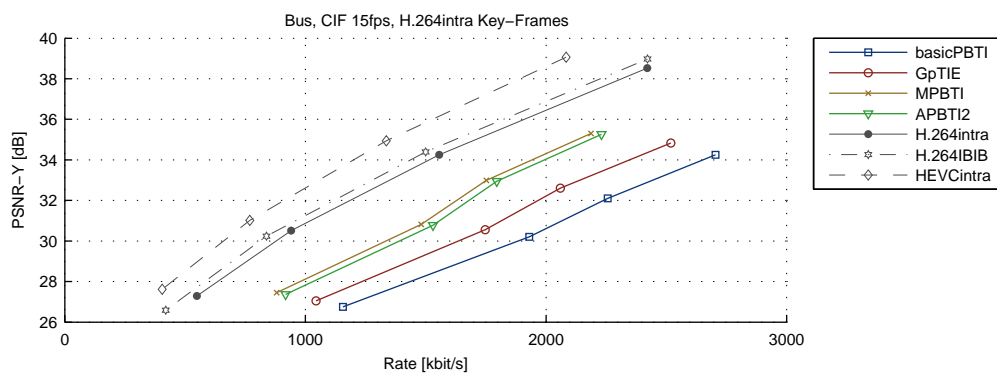


Abbildung D.38.: RD-Performanz des Gesamtcodecs, Bus, CIF, 15 fps

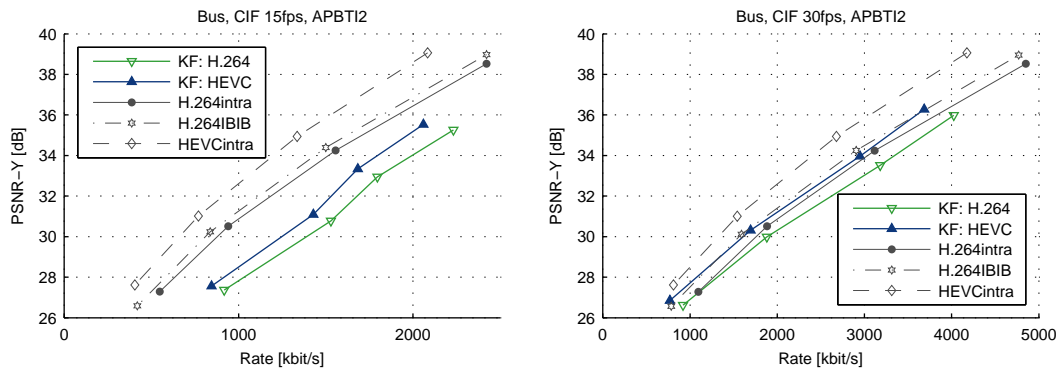


Abbildung D.39.: RD-Performanz des Gesamtcodecs, Bus, CIF

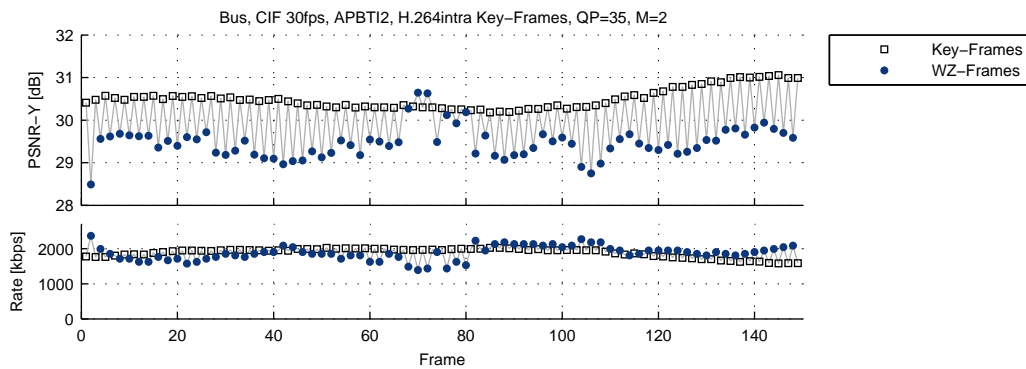


Abbildung D.40.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, Bus, CIF, 30 fps

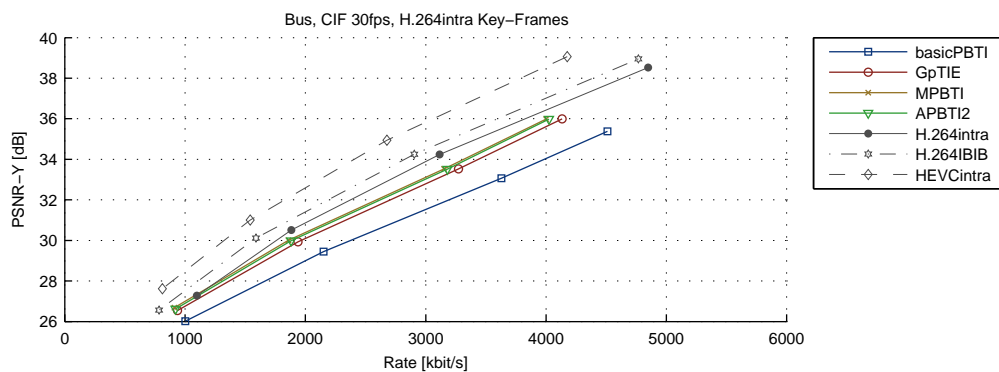


Abbildung D.41.: RD-Performanz des Gesamtcodecs, Bus, CIF, 30 fps

SFlowG – SIF

- die Verwendung von GpTIE zeigt die höchste RD-Performanz
- je nach Framerate mit H.264intra vergleichbare oder bessere Performanz
- teilweise Qualitätseinbrüche der WZ-Frames bis -6 dB, in weiteren Bereichen mit ± 2 dB nur moderate Änderungen

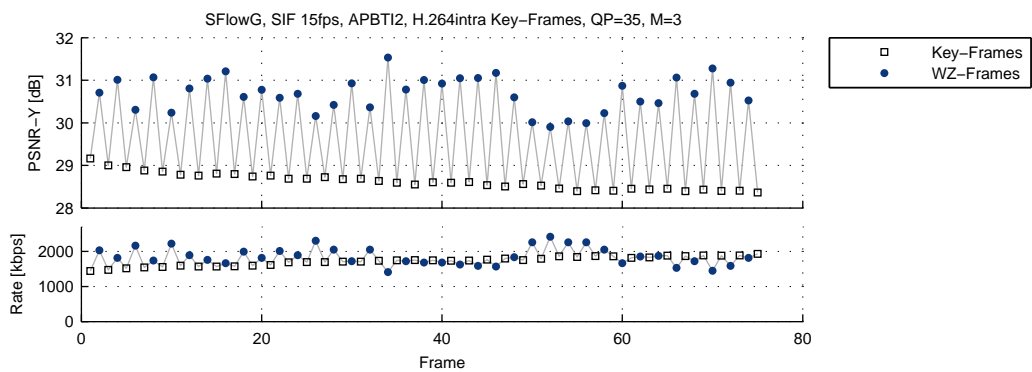


Abbildung D.42.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, SFlowG, SIF, 15 fps

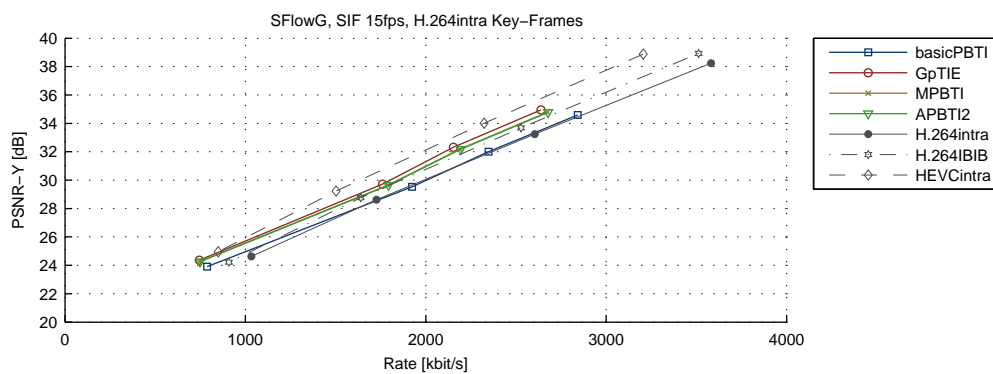


Abbildung D.43.: RD-Performanz des Gesamtcodecs, SFlowG, SIF, 15 fps

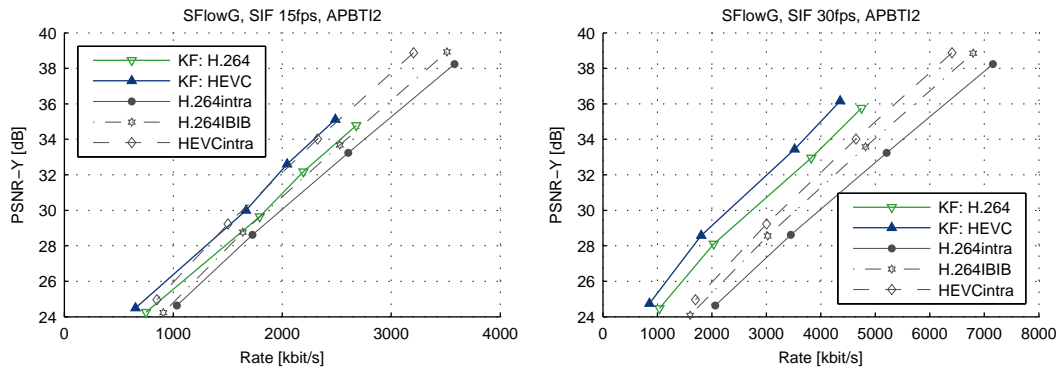


Abbildung D.44.: RD-Performanz des Gesamtcodecs, SFlowG, SIF

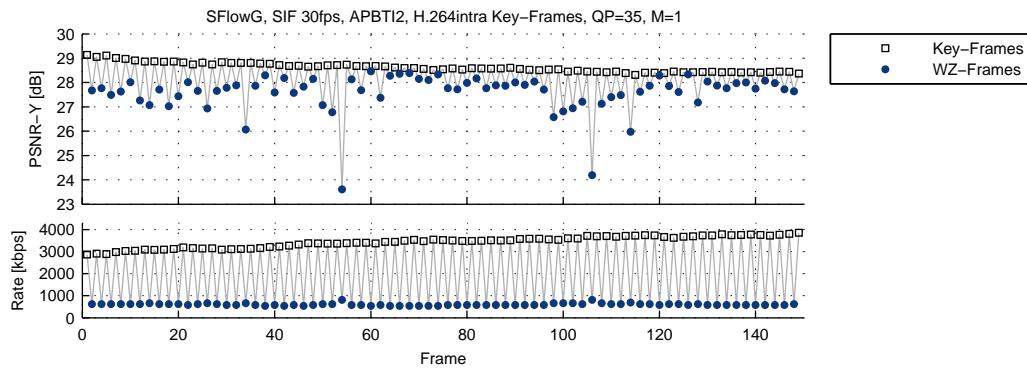


Abbildung D.45.: Qualität und Datenrate der Key- und WZ-Frames, RD-Punkt 2, SFlowG, SIF, 30 fps

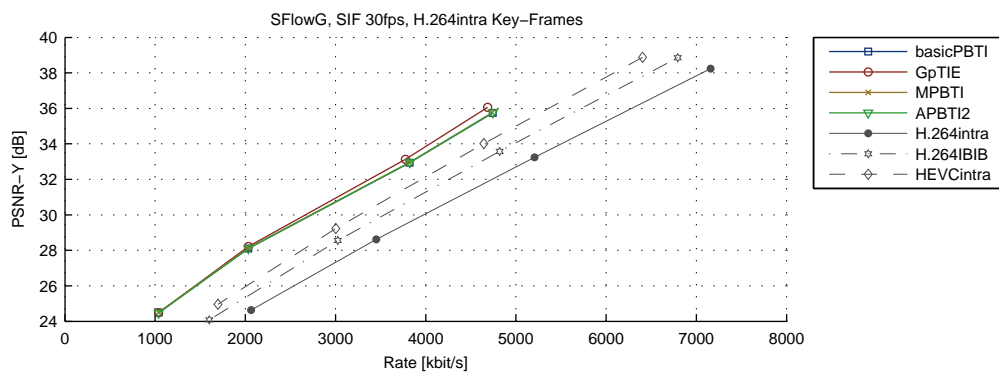


Abbildung D.46.: RD-Performanz des Gesamtcodecs, SFlowG, SIF, 30 fps

Verwendete Rate-Distortion Punkte

RD-Punkt	1	2	3	4
Akiyo – QCIF				
15 fps	(40, 0)	(35, 0)	(30, 0)	(25, 1)
30 fps	(40, 0)	(35, 0)	(30, 0)	(25, 0)
Coastguard – QCIF				
15 fps	(40, 0)	(35, 1)	(30, 2)	(25, 3)
30 fps	(40, 0)	(35, 0)	(30, 0)	(25, 1)
Foreman – QCIF				
15 fps	(40, 1)	(35, 2)	(30, 3)	(25, 3)
30 fps	(40, 0)	(35, 1)	(30, 1)	(25, 2)
Hall – QCIF				
15 fps	(40, 0)	(35, 0)	(30, 1)	(25, 2)
30 fps	(40, 0)	(35, 0)	(30, 0)	(25, 1)
Mother & Daughter – QCIF				
15 fps	(40, 0)	(35, 0)	(30, 0)	(25, 1)
30 fps	(40, 0)	(35, 0)	(30, 0)	(25, 0)
Soccer – QCIF				
15 fps	(40, 3)	(35, 3)	(30, 3)	(25, 3)
30 fps	(40, 2)	(35, 3)	(30, 3)	(25, 3)
Stefan – QCIF				
15 fps	(40, 2)	(35, 3)	(30, 3)	(25, 3)
30 fps	(40, 1)	(35, 2)	(30, 3)	(25, 3)
Bus – CIF				
15 fps	(40, 2)	(35, 3)	(30, 3)	(25, 3)
30 fps	(40, 1)	(35, 2)	(30, 3)	(25, 3)
SFlowG – SIF				
15 fps	(40, 1)	(35, 3)	(30, 3)	(25, 3)
30 fps	(40, 0)	(35, 1)	(30, 3)	(25, 3)

Tabelle D.1.: RD-Punkte für die Bewertung der Leistungsfähigkeit des Gesamtsystems mit H.264intra als auch HEVCintra Key-Frames, Tupel (QP, M) aus dem Quantisierungsparameter QP der Key-Frames und dem Quantisierungsparameter M der WZ-Frames

Verwendete Abkürzungen

<i>oDeViCe</i>	Rostock <u>D</u> istributed <u>V</u> ideo <u>C</u> odec [rodivais]
AIE	<i>a-priori information estimation</i> – Schätzung der a-priori Info.
AIT	<i>a-priori information transmission</i> – Übertragung der a-priori Info.
AQ	<i>adaptive quantisation</i> – adaptive Quantisierung
BCJR	Algorithmus zur Decodierung von Kanalcodes, benannt nach den Autoren: L. <u>B</u> ahl, J. <u>C</u> ocke, F. <u>J</u> elinek und J. <u>R</u> aviv
BiMESS	<i>bidirectional motion estimation and spatial smoothing</i> – bidirektionale Bewegungsschätzung mit örtlicher Glättung
(p)BiME	<i>(pixel-based) bidirectional motion estimation</i> – (Pixelbasierte) bidirektionale Bewegungsschätzung
(p)BME	<i>(pixel-based) backward motion estimation</i> – (Pixelbasierte) Rückwärtsbewegungsschätzung
BPE	<i>bitplane extraction</i> – Bitebenenentrennung
bpp	<i>bit per pixel</i> – Bit pro Pixel
CIF	<i>common intermediate format</i> – Videoformat 352 × 288 px
CRC	<i>cyclic redundancy check</i>
CS	<i>compressive sensing</i>
dB	Dezibel
DCT	<i>discrete cosine transform</i> – diskrete Cosinus-Transformation
DSC	<i>distributed source coding</i> – Verteilte Quellencodierung
DVC	<i>distributed video coding</i> – Verteilte Videocodierung
DWT	<i>discrete wavelet transform</i> – diskrete Wavelet-Transformation
ELC	<i>error locating codes</i> – Codes zur Bestimmung der Fehlerposition
(p)FME	<i>(pixel-based) forward motion estimation</i> – (Pixelbasierte) Vorwärtsbewegungsschätzung
fps	<i>frames per second</i> – Frame pro Sekunde
GME	<i>global motion estimation</i> – globale Bewegungsschätzung

GpTIE	<i>g</i> <u>l</u> <i>o</i> <u>b</u> <i>a</i> <u>l</u> <i>M</i> <u>o</u> <i>t</i> <u>i</u> <i>o</i> <u>n</u> <i>G</i> <u>u</u> <i>d</i> <u>e</u> <i>d</i> <i>p</i> <u>i</u> <i>x<i>e</i><u>l</u>-<i>b</i><u>a</u><i>s</i><u>e</u><i>d</i> <i>a</i><u>d</u><i>a</i><u>p</u><i>t</i><u>i</u><i>v</i><i>e</i> <i>t</i><u>e</u><i>m</i><u>p</u><i>o</i><u>r</u><i>a</i><u>l</u> <i>i</i><u>n</u><i>t</i><u>e</u><i>r</i>- <i>/e</i><u>x</u><i>t</i><u>r</u><i>a<u>p</u><i>o</i><u>l</u><i>a</i><u>t</u><i>i<u>o</u><u>n</u></i></i></i>
GTIE	<i>g</i> <u>l</u> <i>o</i> <u>b</u> <i>a<u>l</u> <i>M</i><u>o</u><i>t</i><u>i</u><i>o<u>n</u> <i>G</i><u>u</u><i>d</i><u>e</u><i>d</i> <i>A</i><u>d</u><i>a<u>p</u><i>t</i><u>i</u><u>v</u><i>e</i> <i>t</i><u>e</u><i>m</i><u>p</u><i>o</i><u>r</u><i>a</i><u>l</u> <i>i</i><u>n</u><i>t</i><u>e</u><i>r</i>-<i>/e</i><u>x</u><i>t</i><u>r</u><i>a<u>p</u><i>o</i><u>l</u><i>a<u>t</u><i>i<u>o</u><u>n</u></i></i></i></i></i></i>
kbps	<i>k</i> <u>i</u> <i>l</i> <u>o</u> <i>b</i> <u>i</u> <i>t <i>p</i><u>e</u><i>r <i>s</i><u>e</u><i>c</i><u>o</u><i>n</i><u>d</u> – Kilobit pro Sekunde</i></i>
LDPC(A)	<i>l</i> <u>o</u> <i>w</i> <i>d</i> <u>e</u> <i>n</i> <i>s</i> <u>i</u> <i>t</i> <u>y</u> <i>p</i> <u>a</u> <i>r</i> <u>i</u> <i>t</i> <u>y</u> <i>c</i> <u>h</u> <i>e</i> <u>c</u> <i>k</i> (<i>a</i> <u>c</u> <i>c<i>u</i><i>m</i><u>u</u><i>l</i><i>a</i><u>t</u><i>e</i>)</i>
LME	<i>l</i> <u>o</u> <i>c</i> <u>a</u> <i>l</i> <i>m</i> <u>o</u> <i>t</i> <u>i</u> <u>o</u> <u>n</u> <i>e</i> <u>s</u> <i>t</i> <u>i</u> <u>m</u> <u>a</u> <i>t</i> <u>i</u> <u>o</u> <i>n</i> – lokale Bewegungsschätzung
LSB	<i>l</i> <u>e</u> <i>a</i> <i>s</i> <i>t</i> <i>s</i> <u>i</u> <i>g</i> <u>n</u> <i>i<i>f</i><u>i</u><i>c</i><u>a</u><i>n</i><i>t</i> <i>b</i><u>i</u><i>t</i> – Bit mit der geringsten Wertigkeit</i>
MAD	<i>m</i> <u>e</u> <i>a</i> <i>n</i> <i>a</i> <u>b</u> <i>s<i>o</i><i>l</i><i>u</i><i>t</i><i>e</i> <i>d</i><u>i</u><i>f<u>f</u><i>e<i>r</i><i>e</i><i>n</i><i>c</i><i>e</i> – Mittelwert der Absolutewerte der Differenzen</i></i></i>
MAP	<i>a</i> <u>a</u> <i>x</i> <i>i<i>m</i><u>u</u><i>m</i>-<i>a</i><u>p</u><i>o</i><i>s</i><i>t</i><u>e</u><i>r</i><u>i</u><u>o</u><i>r</i><i>i (Decodierung)</i></i>
(p)MC	(<i>p</i> <u>i</u> <i>x</i> <u>e</u> <i>l-<i>b</i><u>a</u><i>s</i><u>e</u><i>d</i>) <i>m</i><u>o</u><i>t</i><u>i</u><u>o</u><u>n</u> <i>c</i><u>o</u><i>m</i><u>p</u><i>e<i>n</i><u>s</u><i>a</i><i>t</i><u>i</u><u>o</u><i>n</i> – (Pixelbasierte) Bewegungskompensation</i></i>
MCTI	<i>m</i> <u>o</u> <i>t</i> <u>i</u> <u>o</u> <u>n</u> <i>c</i> <u>o</u> <i>m<u>p</u><i>e<i>n</i><u>s</u><i>a</i><i>t</i><u>e</u><i>d</i> <i>t</i><u>e</u><i>m</i><u>p</u><i>o</i><u>r</u><i>a</i><u>l</u> <i>i</i><u>n</u><i>t</i><u>e</u><i>r</i><u>p</u><i>o</i><u>l</u><i>a<u>t</u><i>i<u>o</u><u>n</u> – bewegungskompensierte zeitliche Interpolation</i></i></i></i>
ML	<i>m</i> <u>a</u> <i>x</i> <u>i</u> <u>m</u> <i>u</i> <i>m</i> - <i>l</i> <u>i</u> <u>k</u> <i>e<i>l</i><i>i<u>h</u><u>o</u><u>o</u><i>d</i> (Decodierung)</i></i>
MPBTI	<i>m</i> <u>u</u> <i>l</i> <u>t</u> <i>i<u>p</u><i>l</i><i>e</i> <i>M</i><u>o</u><i>t</i><u>i</u><u>o</u><u>n</u> <i>H</i><u>y</u><u>p</u><i>o</i><i>t</i><u>h</u><i>e</i><u>s</u><i>e</i>s <i>p</i><u>i</u><i>x<u>e</u><u>l</u>-<i>b</i><u>a</u><i>s</i><u>e</u><i>d</i> <i>t</i><u>e</u><i>m</i><u>p</u><i>o</i><u>r</u><i>a</i><u>l</u> <i>i</i><u>n</u><i>t</i><u>e</u><i>r</i><u>p</u><i>o</i><u>l</u><i>a<u>t</u><i>i<u>o</u><u>n</u> – Multihypothesen pixelbasierte temporale Interpolation</i></i></i></i>
MSB	<i>m</i> <u>o</u> <i>s</i> <i>t</i> <i>s</i> <u>i</u> <i>g<u>n</u><i>i<i>f<u>i</u><i>c<u>a</u><i>n</i><i>t</i> <i>b</i><u>i</u><i>t</i> – Bit mit der höchsten Wertigkeit</i></i></i></i>
MSE	<i>m</i> <u>e</u> <i>a</i> <i>n</i> <i>s</i> <u>q</u> <u>u</u> <i>a</i> <i>r</i> <i>e</i> <i>d</i> <i>e</i> <u>r</u> <i>r</i> <u>o</u> <i>r</i> – mittlerer quadratischer Fehler
NSC	<i>n</i> <u>o</u> <i>n</i> <i>r</i> <i>e</i> <i>c</i> <u>r</u> <i>s</i> <i>i</i> <u>v</u> <i>e</i> <i>n</i> <u>o</u> <i>n</i> <i>s</i> <u>y</u> <i>s</i> <u>t</u> <i>e<i>m</i><u>a</u><i>t</i><i>i<u>c</u> <i>c</i><u>o</u><i>n<i>v</i><u>o</u><i>l<i>u</i><i>t</i><i>i<u>o</u><i>n</i><i>a</i><i>l</i> (<i>c</i><u>o</u><i>d</i><i>e</i>) – nichtrekursiver nichtsystematischer Faltungs-(code)</i></i></i></i></i>
PBTI	<i>p</i> <u>i</u> <i>x</i> <u>e</u> <u>l</u> - <i>b</i> <u>a</u> <i>s</i> <u>e</u> <i>d</i> <i>t</i> <u>e</u> <i>m</i> <u>p</u> <i>o</i> <u>r</u> <i>a</i> <u>l</u> <i>i</i> <u>n</u> <i>t</i> <u>e</u> <i>r</i> <u>p</u> <i>o</i> <u>l</u> <i>a<u>t</u><i>i<u>o</u><u>n</u> – Pixelbasierte zeitliche Interpolation</i></i>
px	Pixel
Q	Quantisierung
QCIF	<i>q</i> <u>u</u> <i>a</i> <i>r</i> <i>t</i> <i>e</i> <i>r</i> <i>C</i> <u>I</u> <i>F</i> – 176×144 px
RSC	<i>r</i> <u>e</u> <i>c</i> <u>r</u> <i>s</i> <i>i<u>v</u><i>e</i> <i>s</i><u>y</u><i>s<u>t</u><i>e<i>m</i><u>a</u><i>t</i><i>i<u>c</u> <i>c</i><u>o</u><i>n<i>v</i><u>o</u><i>l<i>u</i><i>t</i><i>i<u>o</u><i>n</i><i>a</i><i>l</i> (<i>c</i><u>o</u><i>d</i><i>e</i>) – rekursiver systematischer Faltungs-(code)</i></i></i></i></i></i></i>
SAD	<i>s</i> <u>u</u> <i>m</i> <i>o</i> <i>f</i> <i>a</i> <u>b</u> <i>s<i>o</i><i>l</i><i>u</i><i>t</i><i>e</i> <i>d</i><u>i</u><i>f<u>f</u><i>e<i>r</i><i>e</i><i>n</i><i>c</i><i>e</i>s – Summe der Beträge der Differenzen</i></i></i>
SIF	<i>S</i> <u>o</u> <i>u</i> <i>r</i> <i>c</i> <i>e</i> <i>I</i> <u>n</u> <i>p<i>u</i><i>t</i> <i>F</i><u>o</u><i>r<i>m</i><i>a</i><i>t</i> – 352×240 px</i></i>
SW	<i>S</i> <u>l</u> <i>e</i> <i>p</i> <i>i<i>a</i><i>n</i>-<i>W</i><u>o</u><i>l</i><i>f</i> (Codierung)</i>
TCM	<i>t</i> <u>r</u> <i>e<i>l</i><i>l</i><i>i</i><i>s</i>-<i>c</i><u>o</u><i>d</i><i>e</i><i>d</i> <i>m</i><u>o</u><i>d</i><u>u</u><i>l</i><i>a</i><i>t</i><i>i<u>o</u><i>n</i> – Trelliscodemodulation</i></i>
wSAD	<i>w</i> <u>e</u> <i>i<i>g</i><u>h</u><i>t</i><i>e</i><i>d</i> <i>s</i><u>u</u><i>m</i> <i>o</i><i>f</i> <i>a</i><u>b</u><i>s</i><i>o</i><i>l</i><i>u</i><i>t</i><i>e</i> <i>d</i><u>i</u><i>f<u>f</u><i>e<i>r</i><i>e</i><i>n</i><i>c</i><i>e</i>s – gewichtete Summe der Beträge der Differenzen</i></i></i>
WZ	<i>W</i> <u>y</u> <i>n</i> <i>e</i> <i>r</i> - <i>Z</i> <u>i</u> <i>v</i> (Codierung)

Verwendete Formelzeichen

x, N	skalare Variable
M	Menge
\mathbf{X}	Matrix
\mathbf{x}	Zeilenvektor
$\mathbf{X}_{i \times j}$	Matrix bzw. Vektor der Dimension $i \times j$
\mathbf{X}^T	transponierte(r) Matrix bzw. Vektor
x_j	Element eines Vektors
$\mathbf{X}(j, i), x_{i,j}$	Element einer Matrix (i -te Zeile, j -te Spalte)
(x_0, \dots, x_{n-1})	Vektor der Länge n
\mathbf{I}_k	Einheitsmatrix der Größe $k \times k$
\mathbf{G}	Generatormatrix $k \times n$
\mathbf{A}	Generatormatrix ohne systematischen Anteil $(n - k) \times k$
\mathbf{H}	Prüfmatrix $(n - k) \times n$
\mathbb{Q}	rationale Zahlen
\mathbb{Q}_b	rationale Zahlen, die ganzzahliges Vielfache von $\frac{1}{b}$ sind
$\mathbf{X}_i, \mathbf{Y}_i, \mathbf{K}_i$	Wyner-Ziv-Frame, Seiteninformation, Key-Frame
$\mathbf{Y}_{D,i}$	Differenz der bewegungskompensierten benachbarten Key-Frames (<i>dframe</i>)
$\hat{\mathbf{X}}_i$	rekonstruiertes Bild
$\mathbf{K}_i^{\mathbf{MVF}}$	Key-Frame \mathbf{K}_i , welches mit dem Bewegungsvektorfeld \mathbf{MVF} bewegungskompensiert wurde
\mathbf{F}	ein Bild – <i>Frame</i>
\mathbf{GW}_σ	Gaußfenster mit dem Radius σ
$q_i^{(b)}$	b -te Bitebene der Quantisierungssymbols q_i
$\mathbf{q}_Y^{(b)}$	b -te Bitebene der quantisierten Seiteninformation
$\text{UG}(q)$	Untergrenze des q -ten Quantisierungsintervalls
$R^{(b)}$	Datenrate für die Bitebene b

$R_{max}^{(b)}$	maximale Datenrate zur Decodierung der Bitebene b
$R_{min}^{(b)}$	minimale Datenrate zur Decodierung der Bitebene b
R_{suc}	notwendige Gesamtdatenrate zur Decodierung aller Bitebenen
\mathbf{EM}_i	Matrix mit möglichen Fehlerpositionen im Bild (<i>error map</i>)
$\Pr(X = x_\nu)$	Wahrscheinlichkeit, dass die Zufallsvariable X den Wert x_i annimmt
$p_X(x)$	Dichte der kontinuierlichen Zufallsvariablen X
$H(X)$	Entropie der Zufallsvariablen X
$h(X)$	differentielle Entropie der kontinuierlichen Zufallsvariablen X
$I(X = x_i)$	Informationsgehalt des Symbols x_ν der Zufallsvariablen X
$I(X; Y)$	Transinformation zwischen den Zufallsvariablen X und Y
$R(D)$	Rate-Distortion Funktion
$d(x, y)$	Distanz zwischen Skalaren x und y
$[a; b]$	geschlossenes Intervall
$[a; b[$	einseitiges offenes Intervall
$\lfloor x \rfloor$	Abrunden von x auf den nächstkleineren ganzzahligen Wert
$\mathbf{A} \cdot \mathbf{B}$	Matrixmultiplikation
$\mathbf{A} \bullet \mathbf{B}$	Skalarprodukt (auch Matrizen), elementweise Multiplikation and Summation
$\text{Det}(\mathbf{A})$	Determinante der Matrix \mathbf{A}
$\text{Rg}(\mathbf{A})$	Rang der Matrix \mathbf{A}
$\text{Tr}(\mathbf{A})$	Spur (<i>Trace</i> , Summe der Elemente der Hauptdiagonale) der Matrix \mathbf{A}
$(\mathbf{A} \mathbf{B})$	horizontales Aneinanderfügen der Matrizen \mathbf{A} und \mathbf{B}
$\Pi(\mathbf{a}), \Pi^{-1}(\mathbf{a}')$	Interleaving bzw. Deinterleaving des Vektors \mathbf{a}
$\pi(j)$	Index im interleavten Vector \mathbf{a}' , $a'_{\pi(j)} = a_j$
$\arg(\dots)$	Argument einer Funktion, Bsp.: Index i für den x_i minimal ist $i = \arg_i \min(x_i)$

Veröffentlichungen

Hänsel u. Müller 2009a

HÄNSEL, R. ; MÜLLER, E.: Error Locating for plausible Wyner-Ziv Video Coding using Turbo Codes. In: *Proceedings of the 2009 IEEE International Workshop on Multimedia Signal Processing (MMSP 2009)*. Rio de Janeiro, Brazil, 2009. – 6 Seiten, ISBN 978-1-4244-4464-9

Hänsel u. Müller 2009b

HÄNSEL, R. ; MÜLLER, E.: Improved Reconstruction for Distributed Video Coding. In: *Proceedings of the International Conference on Ultra Modern Communications (ICUMT 2009)*. St.-Petersburg, Russia, 2009. – 5 Seiten

Hänsel u. Müller 2009c

HÄNSEL, R. ; MÜLLER, E.: Simplifying the Decoder Rate Control Scheme for Distributed Video Coding by Flexible Slepian-Wolf Decoding. In: *Proceedings of the 3rd Pacific-Rim Symposium on Image and Video Technology (PSIVT 2009)*. Tokyo, Japan, 2009, 1022-1033. – 12 Seiten, ISBN 978-3-540-92956-7

Hänsel u. Müller 2010a

HÄNSEL, R. ; MÜLLER, E.: Confidence Features Extraction for Wyner-Ziv Video Decoding. In: *Proceedings of the 18th European Signal Processing Conference (EUSIPCO 2010)*. Aalborg, Denmark, 2010, 2042-2046. – 5 Seiten, ISSN 2076-1465

Hänsel u. Müller 2010b

HÄNSEL, R. ; MÜLLER, E.: Effect of A Priori Information in Pixel Domain Wyner-Ziv Video Coding. In: *Proceedings of the International Congress on Ultra Modern Communications and Control Systems (ICUMT 2010)*. Moscow, Russia, 2010. – 6 Seiten, ISBN 978-1-4244-7286-4

Hänsel u. Müller 2010c

HÄNSEL, R. ; MÜLLER, E.: Strategien zur Verbesserung der Flexibilität in der verteilten Videocodierung. In: *Tagungsband des 13. Symposiums Maritime Elektrotechnik, Elektronik und Informationstechnik (MarSym 2010)*. Rostock, Germany, 2010. – 6 Seiten

Hänsel u. Müller 2011

HÄNSEL, R. ; MÜLLER, E.: Global Motion Guided Adaptive Temporal Inter-/Extrapolation for Side Information Generation in Distributed Video Coding. In: *Proceedings of 18th IEEE International Conference on Image Processing (ICIP 2011)*. Brussels, Belgium, 2011, S. 2681–2684. – 4 Seiten, ISBN 978-1-4577-1302-6

Hänsel u. a. 2011

HÄNSEL, R. ; RICHTER, H. ; MÜLLER, E.: Incorporating Feature Point-based Motion Hypotheses in Distributed Video Coding. In: *Proceedings of the 3rd International Congress on Ultra Modern Communications and Control Systems (ICUMT 2011)*. Budapest, Hungary, 2011. – 6 Seiten, ISBN 978-963-8111-76-0

Hänsel u. Müller 2012

HÄNSEL, R. ; MÜLLER, E.: Improved Adaptive Temporal Inter-/Extrapolation Schemes for Distributed Video Coding. In: *Proceedings of 29th Picture Coding Symposium (PCS 2012)*. Krakow, Poland, 2012, S. 213–216. – 4 Seiten, ISBN 978-1-4577-2049-9

Sofke u. a. 2009

SOFKE, S. ; HÄNSEL, R. ; MÜLLER, E.: Human Visual System Aware Decoding Strategies for Distributed Video Coding. In: *Proceedings of 27th Picture Coding Symposium (PCS 2009)*. Chicago, USA, 2009. – 4 Seiten

Vorträge

Hänsel 2008

HÄNSEL, R.: *Flexible Slepian-Wolf Decodierung für verteilte Videocodierung*. Aachen : Meeting der ITG-Fachgruppe 3.2 „Digitale Bildcodierung“, Juni 2008. – Dokument ITG32/URo/08_1/198

Hänsel 2009a

HÄNSEL, R.: *Alternative Decoderansätze für die verteilte Videocodierung*. Rostock : Meeting der ITG-Fachgruppe 3.2 „Digitale Bildcodierung“, Juli 2009. – Dokument ITG32/URO/09_1b/214

Hänsel 2009b

HÄNSEL, R.: *Fehlerlokalisierung für die Wyner-Ziv Codierung*. Rostock : 13. Diskussionssitzung der ITG-Fachgruppe 5.1 „Angewandte Informationstheorie“, 2009

Hänsel 2010

HÄNSEL, R.: *Einfluss der Quellenentropie auf Slepian-Wolf Codierung*. Rostock : Meeting der ITG-Fachgruppe 5.1 „Angewandte Informationstheorie“, Oktober 2010

Thesen

1. Die geringe Verfügbarkeit von Ressourcen auf mobilen Endgeräten oder Videosensorknoten führt zu einem Bedarf an Videocodierungssystemen mit geringer Encoderkomplexität.
2. Verfahren aus dem Bereich der Verteilten Videocodierung ermöglichen den Entwurf von Videoencodern mit geringen Ressourcenansprüchen, wobei pixelbasierte Systeme eine besonders geringe Komplexität sowie hohe Flexibilität zeigen.

Generierung der Seiteninformation

3. Die Leistungsfähigkeit (RD-Performanz) eines Verteilten Videocodierungssystems wird stark durch die Qualität der Seiteninformation beeinflusst.
4. Aufgrund der starken zeitlichen Abhängigkeiten in einer Videosequenz bietet sich die Bestimmung der Seiteninformation durch temporale Interpolation oder Extrapolation an, wobei komplexe Bewegungsstrukturen eine Herausforderung darstellen.
5. Schnelle Bewegungen können durch einen zweistufigen temporalen Interpolationsansatz aus robuster globaler Bewegungsschätzung und pixelbasierter Bewegungsschätzung erfasst werden, wodurch eine hohe Seiteninformationsqualität erreicht wird.
6. Multiple Bewegungshypothesen eignen sich zur Unterstützung der Bewegungsschätzung bei inhomogenen Bewegungsstrukturen.

7. Die Bestimmung der Seiteninformation in Aufdeckungs- und Verdeckungsbe-
reichen, an bewegten Objekten und am Bildrand, kann durch adaptive Inter-
polation und Extrapolation verbessert werden.
8. Durch die Kenntnis markanter Bildbereiche wird die Robustheit der Bewe-
gungsschätzung verbessert. Weiterhin kann diese Kenntnis zur Reduktion der
Komplexität genutzt werden.

Wyner-Ziv Codierung

9. Neben der Seiteninformation hat die Wyner-Ziv Codierung starken Einfluss
auf die Leistungsfähigkeit des Gesamtsystems. Das WZ-Codierungsproblem
kann als Kanalcodierungsproblem formuliert und mit der sehr leistungsfähi-
gen Turbo-Codierung umgesetzt werden.
10. Die Art der Terminierung der Turbocodes hat keinen Einfluss auf deren Leis-
tungsfähigkeit, aus Sicht der Slepian-Wolf Codierung.
11. Durch Abstimmung von Interleaver und Punktierungsschema kann eine er-
folgreiche Decodierung der Turbocodes bei der höchsten Datenrate sicherge-
stellt werden.
12. Die Turbo-Codierung erreicht für lange Symbolfolgen eine höhere Leistungs-
fähigkeit, in den für die Verteilte Videocodierung relevanten Bereichen, als
LDPCA Codes.
13. Die Berücksichtigung ungleichmäßig verteilter Eingangssymbole, im Rahmen
der Wyner-Ziv Codierung durch adaptive Quantisierung, führt zu einer star-
ken Verringerung der Leistungsfähigkeit. Die Berücksichtigung der Verteilung
muss im Rahmen der Slepian-Wolf Codierung erfolgen, um die Codierungs-
performanz zu erhöhen.
14. Die Kompressionsverfahren JPEG, JPEG2000, H.264 und HEVC führen zu
Kompressionsartefakten, welche die Generierung der Seiteninformation im
gleichen Maße beeinflussen.

15. Das entwickelte Codierungssystem *qDeViCe* erreicht für eine, mit H.264intra oder HEVCintra vergleichbare, Leistungsfähigkeit, für Testsequenzen mit moderater oder geringer Komplexität der Bewegungsstrukturen.
16. Für Sequenzen mit komplexen Bewegungsstrukturen konnte die Leistungsfähigkeit, insbesondere durch die neu entwickelten Verfahren zur Generierung der Seiteninformation, stark verbessert werden. Eine mit H.264intra vergleichbare RD-Performanz konnte jedoch nicht erreicht werden.

Flexible Decodierung

17. Die Vermeidung und Reduktion der Nutzung des Rückkanals in Verteilten Videocodierungssystemen kann über eine geeignete Encoder- oder Decoder-seitige Ratenkontrolle erfolgen. Eine Erhöhung der Freiheitsgrade des Decoders (flexible Decodierung) erleichtert die Lösung dieser Problemstellung.
18. Die Bitebenen übergreifende Decodierung ermöglicht den Austausch von Informationen zwischen den zu decodierenden Bitebenen und somit die Korrektur ungünstig verteilter Korrekturdaten.
19. Die Soft-Rekonstruktion sowie die Kombination aus unscharfer Fehlerlokalisierung und Fehlerverschleierung stellen Ansätze zur Rekonstruktion bei fehlgeschlagener Slepian-Wolf Decodierung dar.

Lebenslauf

Ralph Hänsel

Persönliche Daten

Geburtsdatum: 12. Juli 1980
Geburtsort: Zerbst
Staatsangehörigkeit: deutsch

Schulische Laufbahn

09/1987 – 07/1993	Goethe-Grundschule, Kyritz
09/1993 – 07/1997	Goethe-Realschule, Kyritz
09/1997 – 06/2000	Gesamtschule mit gymnasialer Oberstufe, Neustadt (Dosse)
06/2000	Erlangung der allgemeinen Hochschulreife

Zivildienst

07/2000 – 05/2001	Zivildienst im Umweltamt des Landkreises Ostprignitz-Ruppin
-------------------	---

Hochschulbildung

10/2001 – 03/2007	Studium Informationstechnik / technische Informatik, Universität Rostock
10/2004	Abschluss als Bachelor of Science
11/2004 – 03/2005	Praktikum Siemens CT IC 2, München, Videocodierung
03/2007	Abschluss als Diplomingenieur für Informationstechnik
04/2007 – 03/2013	Promotionsstudium an der Universität Rostock

Berufliche Laufbahn

06/2001 – 09/2001	Arbeit als Kraftfahrer
04/2007 – 02/2013	wissenschaftlicher Mitarbeiter, Universität Rostock, Institut für Nachrichtentechnik

Selbstständigkeitserklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig angefertigt und keine weiteren, als die angegebenen Quellen verwendet zu haben. Die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen sind als solche gekennzeichnet.

Erklärung über frühere Promotionsversuche

Hiermit erkläre ich, dass ich keine früheren Promotionsversuche unternommen habe.

Rostock, den 23. Oktober 2013

Ralph Hänsel